

*John P. Pagon*

Sol SYSTEMS MANUAL



6200 Hollis Street  
Emeryville, CA. 94608  
Phone: (415) 652-8080

Copyright © 1976, 1977, Processor Technology Corporation  
Third Printing, June, 1977  
Manual No. 106000





May 1961



## PREFACE

This new edition of the Sol Systems Manual contains many revisions and additions. Its release coincides with the release of a new "2708" Personality Module, and the Revision E version of the main circuit board: Sol-PC. The new "Sol-PC Rev E" has several improvements: resistors have been added which increase the reliability of the cassette motor relays, jumper options have been added, and traces moved to improve performance. Many improvements which had been accumulating as update information have been integrated into the text. Section VII, Operating Procedures, and Appendix 5, IC Pin Configurations, are now included. A subsection, Modification for 625 Line Video, has been added. If your copy is missing Section VIII, Theory of Operation, it will be available soon. New divider pages with plastic-coated tabs are included to make it easier to flip to frequently referenced sections.

Much effort has gone towards making this manual complete and accurate. The process of updating and revision always continues, however, and we invite your input. If you should find an error, or have suggestions for improving any of our documentation, please submit your suggestions in writing to our Technical Documentation Department, and they will be given thorough consideration.

The three-ring binder you are holding, is an "easel" binder. The cover is hinged from side to side, as well as down the binding, so that it may form its own "easel" stand. To use this feature, lay the manual open on a table. Bend the full width of the manual along the creased hinge, until a resistance to further bending is felt. Then set the manual up on the table, with the bottom of the pages down against the table, and the top inclining away from you. It is supported from falling by the portion of the binder you have bent back. In this position your hands are free for building, making measurements, or troubleshooting.

The first part of this manual you should read is at the very end: the Updates section. Integrate this information into your manual before you begin.



## CONTENTS OUTLINE

Detailed contents precede each section.

- I INTRODUCTION and GENERAL INFORMATION
- II Sol POWER SUPPLY ASSEMBLY and TEST
- III Sol-PC ASSEMBLY and TEST
- IV PERSONALITY MODULE ASSEMBLY
- V KEYBOARD ASSEMBLY and TEST
- VI Sol CABINET-CHASSIS ASSEMBLY
- VII OPERATING PROCEDURES
- VIII THEORY OF OPERATION
- IX SOFTWARE
- X DRAWINGS
- APPENDICES
- UPDATES





## PROCESSOR TECHNOLOGY CORPORATION

Sol TERMINAL COMPUTER<sup>TM</sup>

## LIST OF ILLUSTRATIONS

<u>FIGURE</u>	<u>TITLE</u>	<u>PAGE</u>
2-1	Sol-20 fan closure plate assembly . . . . .	II-7
2-2	Coaxial cable preparation . . . . .	II-9
2-3	Aluminum heat sink installation . . . . .	II-12
2-4	Sol-10 power supply subchassis assembly . . . . .	II-15
2-5	Sol-20 power supply subchassis assembly . . . . .	II-16
2-6	Sol-PC power connector and voltage measurements . . . . .	II-19
2-7	Sol-20 power connector and voltage measurements . . . . .	II-19
3-1	Identification of components . . . . .	III-5
3-2	Clock circuit waveforms . . . . .	III-15
3-3	Test probe for Steps 16B and 25B . . . . .	III-16
3-4	Coaxial cable preparation . . . . .	III-19
3-5	Display section timing waveforms . . . . .	III-21
3-6	Bending selected pins on U42, 59 and 75 . . . . .	III-23
3-7	U14 through U21 socket jumpers . . . . .	III-24
3-8	Display circuits test pattern . . . . .	III-25
3-9	CPU Functional Test No. 1 display . . . . .	III-29
3-10	CPU Functional Test No. 2 display . . . . .	III-31
3-11	Personality module bracket/guide installation . . . . .	III-31
4-1	R1 through R4 installation . . . . .	IV-4
4-2	Handle bracket (Sol-1045) installation . . . . .	IV-5
6-1	Types of screws used in Sol cabinet-chassis assembly . . . . .	VI-4
6-2	Brackets used in Sol cabinet-chassis assembly . . . . .	VI-4
6-3	Sol-20 with covers removed . . . . .	VI-11
6-4	Sol-20 with covers removed . . . . .	VI-11
6-5	Sol-PC coaxial cable connector assembly . . . . .	VI-14
6-6	Backplane board (Sol-BPB) installation . . . . .	VI-16
6-7	Backplane board (Sol-BPB) installation . . . . .	VI-17
6-8	Protective foot pad installation . . . . .	VI-20
7-1	Connecting the basic Sol system . . . . .	VII-6
7-2	Sol control switch settings for terminal mode . . . . .	VII-7
7-3	Location of positioning adjustments, VR1 and VR2 . . . . .	VII-8

## PROCESSOR TECHNOLOGY CORPORATION

Sol TERMINAL COMPUTER<sup>TM</sup>

ILLUSTRATIONS/TABLES

<u>FIGURE</u>	<u>TITLE</u>	<u>PAGE</u>
7-5	Connecting Sol to two cassette recorders . . . . .	VII-29
7-6	Connecting Sol SDI to current loop device such as TTY	VII-31
7-7	Connecting Sol SDI to communications modem . . . . .	VII-31
7-8	Connecting Sol PDI to parallel device . . . . .	VII-32

<u>TABLE</u>	<u>TITLE</u>	<u>PAGE</u>
2-1	Sol Regulator Parts List . . . . .	II-2,3
2-2	Sol-10 Power Supply Parts List . . . . .	II-4
2-3	Sol-20 Power Supply Parts List . . . . .	II-4
3-1	Sol-PC Parts List . . . . .	III-2,3,4
4-1	PM2708 Personality Module Parts List . . . . .	IV-1
5-1	Sol Keyboard Parts List . . . . .	V-2,3
6-1	Sol-10 Cabinet-Chassis Parts List . . . . .	VI-2
6-2	Sol-20 Cabinet-Chassis Parts List . . . . .	VI-3
7-1	Sol Operating Controls and Their Functions . . . . .	VII-2
7-2	Baud Rate Selection with Switch S3 . . . . .	VII-15
7-3	Word Length Selection with S4-2 & 3 . . . . .	VII-15
7-4	Sol Keyboard Assignments . . . . .	VII-18
7-5	Control Character Symbols and Definitions . . . . .	VII-23



## I INTRODUCTION and GENERAL INFORMATION

1.1	Introduction . . . . .	I-1
1.2	General Information . . . . .	I-1
1.2.1	Sol-PC Description . . . . .	I-1
1.2.2	Receiving Inspection . . . . .	I-2
1.2.3	Warranty Information . . . . .	I-2
1.2.4	Replacement Parts . . . . .	I-2
1.2.5	Factory Service . . . . .	I-3





## 1.1 INTRODUCTION

This manual supplies the information needed to assemble, test and use the Sol-PC Single Board Terminal Computer. We suggest that you first scan the entire manual before starting assembly. Then make sure you have all the parts and components listed in the "Parts List" (Table 3-1) in Section III. When assembling the module, follow the instructions in the order given.

Should you encounter any problem during assembly, call on us for help if necessary. If your completed module does not work properly, recheck your assembly step by step. Most problems stem from poor soldering, backward installed components, and/or installing the wrong component. Once you are satisfied that the module is correctly assembled, feel free to ask for our help.

## 1.2 GENERAL INFORMATION

### 1.2.1 Sol-PC Description

The Sol-PC is a single board microcomputer/terminal built around an 8080 microprocessor. Support circuitry permits full implementation of every 8080 function.

It features both parallel and serial communications interfaces, a keyboard interface, an audio cassette interface, a video display generator, 1024 8-bit words of system RAM (random access memory), 1024 8-bit words of display RAM, and a plug-in personality module with up to 2048 bytes of ROM (read only memory) stored program, and bus compatibility with all Processor Technology hardware and firmware products. Power requirements for the Sol-PC are +5 V dc  $\pm 5\%$  at 2.5 A, +12 V dc  $\pm 5\%$  at 150 mA and -12 V dc  $\pm 5\%$  at 200 mA.

Parallel interfacing is eight bits each for input and output plus control handshaking signals, and the output bus is tristated TTL for bidirectional interfaces. The serial interface circuit includes both asynchronous RS-232 and 20 mA current loop provisions, 75 to 9600 baud (switch selectable).

Seven-level ASCII encoded, TTL keyboard interfacing requires a 2 to 10  $\mu$ sec strobe pulse after data is stable. The dual rate, 300 or 1200 bps (bits per second), audio cassette interface is program controlled and self clocking with phase-lock loop. It includes automatic level control in both the record and playback modes. Recording is CUTS/Byte standard compatible, asynchronously Manchester coded at 1200/2400 Hz or 600/1200 Hz.

The video display circuitry generates sixteen 64 character lines from data stored in an on-card 1024 8-bit word display RAM. Alphanumeric and control characters (the full 128 upper and lower case plus control ASCII character set) are displayed black on white

or reverse (switch selectable). Solid video inversion cursors, with switch selectable blink, are programmable. The display output is standard EIA, 1.0 to 2.5 V p-p with composite negative sync, with a nominal bandwidth of 7 MHz. It can consequently be used to drive any standard video monitor. (A monochrome TV, converted for video input, can also be used. See Appendix VI.)

Included on the card are 1024 words of static, low power system RAM capable of full speed operation and a plug-in personality module which contains the software control program. Three personality modules are available for Sol:

CONSOL<sup>TM</sup>--allows simple terminal operations plus direct control of the basic computer functions for entering or examining data in any memory location, or executing a program stored at a known location in memory.

SOLED<sup>TM</sup>--allows advanced terminal operations with CONSOL plus screen, file and cassette tape editing/transmission operations.

SOLOS<sup>TM</sup>--allows full stand-alone terminal-computer operation.

### 1.2.2 Receiving Inspection

When your kit arrives, examine the shipping container for signs of possible damage to the contents during transit. Then inspect the contents for damage. (We suggest you save the shipping materials for use in returning the kit to Processor Technology should it become necessary to do so.) If your Sol-PC kit is damaged, please write us at once describing the condition so that we can take appropriate action.

### 1.2.3 Warranty Information

In brief, parts which fail because of defects in materials or workmanship are replaced at no charge for 3 months for kits, and one year for assembled products, following the date of purchase. Also, products assembled by the buyer are warranted for a period of 3 months after the date of purchase; factory assembled units carry a one year warranty. Refer to Appendix I for the complete "Statement of Warranty".

### 1.2.4 Replacement Parts

Order replacement parts by component nomenclature (DM8131 IC or 1N2222 diode, for example) and/or a complete description (680 ohm, 1/4 watt, 5% carbon resistor, for example).



## 1.2.5 Factory Service

In addition to in-warranty service, Processor Technology also provides factory repair service on out-of-warranty Processor Technology products. Before returning the unit to us, first obtain our authorization to do so by writing us a letter describing the problem. After you receive our authorization to return the unit, proceed as follows:

1. Write a description of the problem.
2. Pack the unit with the description in a container suitable to the method of shipment.
3. Ship prepaid to Processor Technology Corporation, 6200 Hollis Street, Emeryville, CA 94608.

Your unit will be repaired as soon as possible after receipt and return shipped to you prepaid. (Factory service charges will not exceed \$20.00 without prior notification and your approval.)



## II Sol POWER SUPPLY ASSEMBLY and TEST

2.1	Introduction . . . . .	II-1
2.2	Parts and Components . . . . .	II-1
2.2.1	Sol Regulator (Sol-REG) . . . . .	II-1
2.2.2	Power Supply Subchassis and Components . . . . .	II-1
2.3	Assembly Tips . . . . .	II-5
2.3.1	Electrical . . . . .	II-5
2.3.2	Mechanical . . . . .	II-5
2.4	Assembly Precautions . . . . .	II-5
2.5	Required Tools, Equipment and Materials . .	II-6
2.6	Orientation . . . . .	II-6
2.6.1	Sol-REG PC Board . . . . .	II-6
2.6.2	Fan Closure Plate . . . . .	II-6
2.7	Assembly-Test . . . . .	II-6
2.7.1	Fan Closure Plate Assembly . . . . .	II-7
2.7.2	Sol-REG Assembly and Test . . . . .	II-10
2.7.3	Power Supply Subchassis Assembly and Test . . . . .	II-14





## 2.1 INTRODUCTION

The Sol power supply consists of a regulator board plus additional chassis-mounted components. This section covers assembly and test of the complete power supply.

## 2.2 PARTS AND COMPONENTS

### 2.2.1 Sol Regulator (Sol-REG)

Check all parts and components against the appropriate "Parts List", Tables 2-1, 2-2 and 2-3. If you have difficulty in identifying any parts by sight, refer to Figure 3-1 on Page III-5 in Section III of this manual.

### 2.2.2 Power Supply Subchassis and Components

In addition to the Sol-REG, you will need the following parts and components supplied with the Sol Cabinet-Chassis Kit. Check these parts against the appropriate "Parts List(s)", Tables 6-1 and 6-2, in Section VI and separate them from the other cabinet-chassis parts.

Fan Closure Plate

Power Supply Subchassis (L-shaped)

4 each 4-40 x 3/16 Machine Screw

4 each 4-40 x 5/16 Machine Screw

4 each 4-40 Hex Nut

10 each #4 Lockwasher

14 each 6-32 x 1/2 Machine Screw

14 each 6-32 Hex Nut

16 each #6 Lockwasher

3 each 8-32 x 1/2 Machine Screw

3 each 8-32 Hex Nut

3 each #8 Lockwasher

11 each #6 x 1/4 Sheet Metal Screw

1 each #6 x 5/16 Sheet Metal Screw

2 each #4 Solder Lug

2 each 1/4" Spacer, 4-40 Tapped



Table 2-1. Sol Regulator Parts List.

<u>INTEGRATED CIRCUITS**</u> 1 1458 (U2) 1 7812 (U1) 1 7912 (U3)	<u>DIODES and RECTIFIERS</u> 1 MDA101A (FWB2) 1 MDA970-1 (FWB1) 1 IR106B2 or MCR106-2 (SCR1) 2 1N4001 (D3 & 4) 1 1N4148 (D2) 1 1N5231B (D1)
<u>TRANSISTORS</u> 2 2N2222 (Q2 & 3) 1 T1P41 (Q1)	
<u>RESISTORS</u> 1 0.1 ohm, 3 watt, 5% or 5 watt, 5% 1 68 ohm, ¼ watt, 5% 1 330 ohm, ¼ watt, 5% 2 1 K ohm, ¼ watt, 5% 4 10 K ohm, ¼ watt, 5% 1 56 K ohm, ¼ watt, 5% 1 1690 ohm, ¼ watt, 5% 1 4020 ohm, ¼ watt, 5%	<u>CAPACITORS</u> 2 .1 ufd, disc 3 15 ufd, tantalum dipped 2 2500 ufd, tubular electrolytic 1 *18,000 ufd, electrolytic
<u>CABLE ASSEMBLIES</u> 1 *Single wire, 3" (Fuse Holder to Power Switch) 1 *Single wire, 3¼" (Power Switch to Commoning Block) 1 Two wire, 10" (C8 to Regulator Board)	

\*Chassis-mounted component

\*\*When identifying IC's, you can ignore prefix and suffix characters in the IC nomenclature since these vary with the manufacturer. For example a 1458CP, 1458CPI and MCL458N are all 1458 IC's. This applies to all Parts Lists in this manual.

Table 2-1. Sol Regulator Parts List (Continued).

MISCELLANEOUS

- 1 Sol REG Circuit Board
- 1 Heat Sink, 690-220-P
- 1 Heat Sink, 203-AP
- 1 Heat Sink, aluminum
- 1 Package Heat Sink Compound
- 2 Coax Connector, female\* (Video Output)
- 1 Coax Connector, male (Video Output Cable)
- 1 Coax Connector Adapter Sleeve (Video Output Cable)
- 1 \*AC Receptacle, female
- 1 \*Fuse Holder
- 1 \*SPST Power Switch, pushbutton (S5)
- 1 AC Power Cord
- 2 \*Commoning Blocks
- 1 \*Clamp for C8, 1½"
- 4 Tie Wraps
- 3 Mica Insulators
- 1 4-40 x 7/16 screw
- 1 4-40 x 5/8 screw
- 2 4-40 Hex Nut
- 1 6-32 x ½ screw, metal
- 2 6-32 x ½ screw, Nylon
- 3 6-32 Hex Nut
- 5 #4 Lockwasher, internal tooth
- 1 Length Solder

\*Chassis-mounted component

Table 2-2. Sol-10 Power Supply Parts List.

The Sol-10 Power Supply Kit includes all Sol-REG parts listed in Table 2-1 plus the following components:	
<hr/>	
1 *Power Transformer, T1	
1 *Fuse, 3 amp Slo-Blo (F1)	

\*Chassis-mounted component

Table 2-3. Sol-20 Power Supply Parts List.

The Sol-20 Power Supply Kit includes all Sol-REG parts listed in Table 2-1 plus the following components:	
<hr/>	
<u>RESISTORS</u>	<u>CAPACITORS</u>
1 *39 ohm, 2 watt, 5%	1 *54,000 ufd, electrolytic
<u>RECTIFIERS</u>	<u>TRANSFORMERS</u>
1 *MDA980-1 (FWB3)	1 *Power Transformer, T2
<u>MISCELLANEOUS</u>	
1 *Fan	1 5-wire Cable Assembly
1 *Fan Guard	1 *Clamp for C9, 2½"
1 *Fuse, 3 amp Slo-Blo	2 *#10 solder lug, internal tooth

\*Chassis-mounted component

## 2.3 ASSEMBLY TIPS

### 2.3.1 Electrical

For the most part the assembly tips given in Paragraph 3.2 of Section III (Page III-1) apply to assembling the Sol regulator board and power supply.

In addition, scan Section II completely before you start to assemble the power supply.

### 2.3.2 Mechanical

1. If you do not have the proper screwdrivers (see Paragraph 2.5), we recommend that you buy them rather than using a knife point, a blade screwdriver on a Phillips screw, and other makeshift means. Proper screwdrivers minimize the chances of stripping threads, disfiguring screw heads and marring decorative surfaces.

2. To assure a correct fit and tight assembly, be sure you use the screws specified in the instructions.

3. Lockwashers are widely used in the power supply assembly so that screws will not loosen when subjected to stress or vibration. When a lockwasher is specified, do not omit it and make sure you install it correctly.

4. Some instructions call for prethreading holes. This is done to make assembly easier by giving you maximum working space for installing relatively hard-to-drive sheet metal screws. If you bypass prethreading instructions you will only make subsequent cabinet-chassis assembly more difficult.

To prethread a hole, insert specified screw in the hole and position it as straight as possible. While holding the screw in this position, drive it into the metal with the proper screwdriver. If started straight the screw will continue to go straight into the metal so that the head and sheet metal surfaces are in full contact.

5. The diameter of the shank (threaded portion) of a screw increases in relation to its number. For example, a 6-32 screw is larger in diameter than a 4-40 screw. Also, a #8 lockwasher is larger than a #4 lockwasher.

6. Heat sink compound is supplied with this kit in a small clear plastic package. It is a thick white substance which improves heat transfer between components and their heat sinks. To use the compound, pierce a small hole near the edge of the top surface of the plastic package, using a pin or sharp knife point. Squeezing the package will cause a small amount of the compound to ooze out

## Sol POWER SUPPLY

## SECTION II

out of the hole, which may then be applied with a toothpick or small screwdriver blade. Spread a thin film of the compound on the mating surfaces of both the heat-generating component and the heat sink surface which it will contact. Then assemble as directed.

## 2.4 ASSEMBLY PRECAUTIONS

The precautions concerning soldering and the installation and removal of integrated circuits given in Paragraph 3.3 of Section III (Page III-6) also apply to assembling the Sol regulator board.

## 2.5 REQUIRED TOOLS, EQUIPMENT AND MATERIALS

The following tools, equipment and materials are recommended for assembling the Sol regulator board:

1. Needle nose pliers
2. Diagonal cutters
3. Sharp knife
4. Screwdriver, thin  $\frac{1}{4}$ " blade
5. Screwdriver, #2 Phillips
6. Controlled heat soldering iron, 25 watt
7. 60-40 rosin-core solder (supplied)
8. Volt-ohm meter
9. Ruler

## 2.6 ORIENTATION

## 2.6.1 Sol-REG PC Board

Location C5 (2500 ufd capacitor) will be located in the lower right-hand corner of the circuit board when locations SCR1, Q1 and FWB1 are positioned along the top of the board. In this position the component (front) side of the board is facing up and the horizontal legends will read from left to right; the other legends will read from bottom to top. Subsequent position references related to the Sol-REG board assume this orientation.

## 2.6.2 Fan Closure Plate

The large circular cutout will be located in the upper right quadrant of the plate when the heavy gauge doubler plate is facing up. In this position the rectangular cutouts are on the left, the front side of the plate is facing down, the back side is facing up, and the small circular cutout is at the bottom. We suggest you label the two sides.

## 2.7 ASSEMBLY-TEST

NOTE: Instructions that apply only to the Sol-20 are preceded by an asterisk. Skip these instructions if you are assembling a Sol-10.

## 2.7.1 Fan Closure Plate Assembly

Refer to Assembly Drawings on Pages X-1 and 2 in Section X.  
(Figure 2-1 shows a completed fan closure plate assembly.)

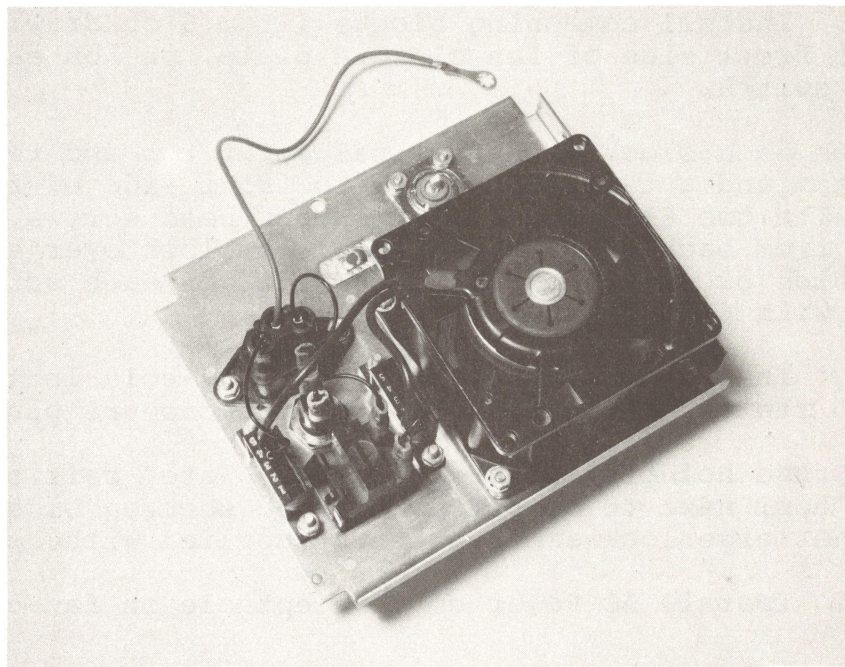


Figure 2-1. Sol-20 fan closure plate assembly.  
(Top of plate in foreground.)

- \*( ) Step 1. Mount cooling fan and guard to fan closure plate.

Insert four 6-32 x  $\frac{1}{2}$ " binder or pan head screws from back side of fan closure plate. (Use the holes positioned in each quadrant of the large circular cutout.) Slip fan guard over screws on front side of plate. Position fan so that air flow will be from front to back side of plate and with its leads next to the rectangular cutouts in the plate. Place #6 lockwasher on each screw and secure with 6-32 hex nut.

WARNING

FAILURE TO INSTALL FAN GUARD MAY RESULT  
IN DAMAGE TO THE Sol AND/OR PERSONAL  
INJURY.

- ( ) Step 2. Install power on-off switch in upper rectangular cutout in fan closure plate.

(Step 2 continued on Page II-8.)

## Sol POWER SUPPLY

## SECTION II

Bend four retainer tabs on switch in and position switch with terminals facing front side of fan closure plate. Push switch unit from back side of plate through mounting hole and bend retainer tabs outward if needed to hold switch in place.

- ( ) Step 3. Install commoning blocks (Item 6 on drawing on Page X-1) on front side of fan closure plate, one on each side of on-off switch.

Position each block with terminal #1 at top and terminal #5 at bottom and attach each block to front side of fan closure plate with two 6-32 X  $\frac{1}{2}$  binder or pan head screws. Insert screws from back side of plate, place block over screws, on front side of plate, put #6 lockwasher on each screw and secure with 6-32 hex nut.

- ( ) Step 4. Install fuse holder in mounting hole located between the two rectangular cutouts in the fan closure plate.

Insert fuse holder from back side of plate, position large tab at top, next to on-off switch, and secure holder to plate with the large lockwasher and nut supplied with holder.

- ( ) Step 5. Install AC Power cord receptacle on fan closure plate.

Position receptacle on front side of fan closure plate over the rectangular cutout below fuse holder. Orient receptacle with green lead at the bottom and align the receptacle and closure plate mounting holes. Insert two 6-32 x  $\frac{1}{2}$  binder or pan head screws from back side of plate through each mounting hole, put #6 lockwasher on each screw and secure with 6-32 hex nut. Be sure receptacle is properly seated in cutout before tightening to avoid damage.

- ( ) Step 6. Install female coaxial connector on fan closure plate.

Insert connector from front side of plate so that the threaded end projects through to the back side. Then insert four 4-40 x  $\frac{5}{16}$  binder or pan head screws from back side of plate through the four connector and plate mounting holes. Place #4 lockwasher on each screw except the upper one which is closest to the AC receptacle. Secure with 4-40 hex nuts. (Leave upper nut closest to receptacle loose.)

- ( ) Step 7. Prepare RG59/U coaxial cable.

Cut a 13" piece of coaxial cable from that supplied with the Sol-PC kit. Strip away one inch of the outer insulation at both ends to expose shield. Unbraid shield at one end and twist it into a single lead. Do the same thing at the other end. Tin shield lead at each end and solder a #4 lug to each lead. Then remove  $\frac{1}{2}$ " of the inner conductor insulation at both ends. (See Figure 2-2.)

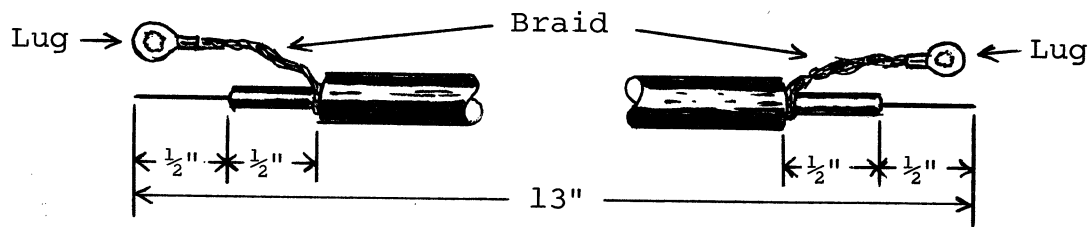


Figure 2-2. Coaxial cable preparation.

- ( ) Step 8. Connect coaxial cable to coaxial connector installed in Step 6.

Solder inner conductor on one end to the pin of the connector. Remove hex nut on upper connector mounting screw closest to AC receptacle, place lug (coaxial shield) on screw and reinstall hex nut.

- ( ) Step 9. Connect fan closure plate wiring.

( ) Install the 3" power switch-to-commoning block cable supplied with your Sol-REG kit. Connect the female spade lug end to the upper terminal of the on-off switch and the commoning block lug end to the #1 terminal of the commoning block closest to the fan. NOTE: To install commoning block lugs, position lug with its open side facing away from the terminal numbers on the block. Then gently push lug into appropriate terminal receptacle until it is fully seated.

( ) Install the 3 1/4" fuse holder-to-power switch cable supplied with your Sol-REG kit. (This cable has female spade lugs at both ends.) Connect one end to the bottom terminal of the on-off switch and the other to the longer male spade lug on the fuse holder.

( ) Connect the AC receptacle wire closest to the fan to the other fuse holder lug. NOTE: The green AC receptacle wire will be connected later.

( ) Connect other AC receptacle wire to terminal #4 on the commoning block furthest away from the fan.

\* ( ) Connect upper wire of fan cord to terminal #3 of the commoning block closest to fan.

\* ( ) Connect lower wire of fan cord to terminal #5 of commoning block furthest from fan.

( ) Put fan closure assembly aside.



## Sol POWER SUPPLY

## SECTION II

## 2.7.2 Sol-REG Assembly and Test

Circuit references, values and outlines are printed on the component side of the board to assist in assembly.

- ( ) Step 10. Visually check Sol-REG board for solder bridges (shorts) between traces, broken traces and similar defects.

If visual inspection reveals any defects, return the board to Processor Technology for replacement. If the board is not defective, proceed to next paragraph.

- ( ) Step 11. Install the following resistors in the indicated locations. Bend leads to fit distance between mounting holes, insert leads, pull down snug to board, solder and trim.

<u>LOCATION</u>	<u>VALUE (ohms)</u>	<u>COLOR CODE</u>
( ) R1	.1, 3 watt	none
( ) R2	330 , 5 watt	orange-orange-brown
( ) R3	10 K	brown-black-orange
( ) R4	10 K	" " "
( ) R5	1 K	brown-black-red
( ) R6	68	blue-gray-black
( ) R7	10 K	brown-black-orange
( ) R8	1 K	brown-black-red
( ) R9	56 K	green-blue-orange
( ) R10	10 K	brown-black-orange
( ) R11	1690	bronw-blue-white-brown
( ) R12	4020	yellow-black-red-brown

- ( ) Step 12. Install U2 (1458) in its location between C2 and C3. U2 is positioned with pin 1 in the lower left-hand corner and soldered into place. See "Loading DIP Devices" in Appendix IV.

- ( ) Step 13. Install diodes D1 (1N5231B), D2 (1N4148), D3 and D4 (1N4001). Bend leads to fit distance between mounting holes, insert leads, pull down snug to board, solder and trim. BE SURE to position D1 with its cathode (dark band) to the left, D2 and D3 with their cathode at the bottom, and D4 with its cathode at the top.

- ( ) Step 14. Install the following capacitors in the indicated locations. Take care to observe the proper value, type and orientation, if applicable, for each installation. Bend leads outward on solder (back) side of board, solder and trim.

(See NOTE on Page II-11.

NOTE

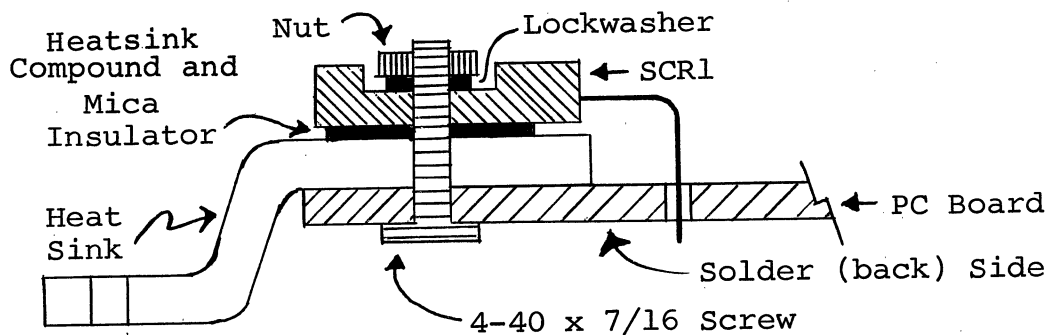
Disc capacitor leads are usually coated with wax during the manufacturing process. After inserting leads through mounting holes, remove capacitor and clear the holes of any wax. Reinsert and install.

<u>LOCATION</u>	<u>VALUE (ufd)</u>	<u>TYPE</u>	<u>ORIENTATION</u>
C1	15	Tantalum	"+" lead bottom right
C2	.1	Disc	None
C3	.1	Disc	None
C6	15	Tantalum	"+" lead right
C7	15	Tantalum	"+" lead left

- ( ) Step 15. Install 2500 ufd capacitors in locations C4 and C5. Bend leads to fit distance between mounting holes, insert leads, pull down snug to board, solder and trim. Be sure to install C4 with its "+" lead to the right and C5 with its "+" lead to the left.
- (X) Step 16. Install Q2 and Q3 (2N2222) in their locations. The emitter lead (closest to tab on can) of Q2 is oriented toward the left and the base lead toward the bottom. The emitter lead of Q3 is oriented toward the bottom and the base lead toward the right.
- ( ) Step 17. Read assembly tip 6, on page II-5. Apply heat sink compound to the inside of the small black "star-shaped" cooling fin, and install it, with the cylindrical grip down, on Q2 by slipping it down onto the can. Be sure heat sink does not touch any other component on the board.
- (X) Step 18. Install bridge rectifier FWB 2 (MDA101A) in its location at the bottom of the board. Apply heat sink compound, per Assembly tip 6 on page II-5. Position FWB2 with its "+" lead at the top and its "-" lead at the bottom, insert leads, solder and trim.
- ( ) Step 19. Install large heat sink, U1 and U3 in their locations on the bottom left corner of the circuit board.
- ( ) Position large black heat sink, (flat side to board) over the square foil area in the lower left corner of the PC board. Orient sink so that the two triangular cutouts in the sink are over the two triangles of mounting holes in the board.
- ( ) Position U1 (7812) on heat sink and observe how leads must be bent to fit mounting holes. Note that the center lead must be bent down approximately 0.2 inches.

further from the body than the other two leads. Bend leads so that no contact is made with the heat sink when U1 is flat against the sink and its mounting hole is aligned with the holes in the sink and PC board. Apply heat sink compound per Assembly Tip 6, on page II-5. Fasten U1 and sink to board using a 6-32 x  $\frac{1}{2}$  metal screw, lockwasher and nut. Insert screw from back (solder) side of board and drive nut finger tight.

- ( ) Position U3 (7912) on heat sink, determine how leads must be bent as you did for U1, and bend leads. Place a rectangular mica insulator over the leads of U3 so that it fully covers the bottom side of the U3 package. Apply heat sink compound to U3, the heat sink, and both sides of the mica insulator. Bend the two outside leads of U3 slightly in toward the center lead, insert leads in mounting holes as you did for U1, and fasten U3 to heat sink and PC board using a 6-32 x  $\frac{1}{2}$  Nylon screw, lockwasher and nut. Insert screw from back (solder) side of board and drive nut finger tight.
- ( ) Position heat sink, U1 and U3 as needed to obtain correct fit and tighten the U1 and U3 mounting screws. REMEMBER, NO LEADS CAN CONTACT THE SINK. Solder all leads and trim if required.
- ( ) Step 20. Install aluminum heat sink, SCR1, Q1 and bridge rectifier FWB1.
- ( ) Position aluminum heat sink (see Figure 2-3) along top of PC board so that the three holes in one side of the sink are aligned with the SCR1, Q1 and FWB1 mounting holes in the PC board.



(Left end, cross-section view)

Figure 2-3. Aluminum heat sink installation.

## ASSEMBLY PROCEDURE CHANGE NOTICE #6-2 Rev B

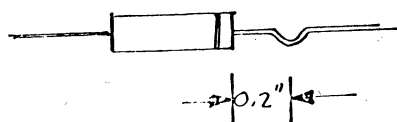
This Change Notice concerns the Sol-REG board and applies only to Revision Level B boards.

A problem was detected in early Sol-REG boards in which the "crow-bar" circuit would trigger without adequate cause and short circuit the 5-volt output. A circuit change has been made which will be reflected in Revision Level C and above boards to correct the problem. Revision Level B boards, however, require the following modification to correct the problem. Parts for this modification are supplied with your kit:

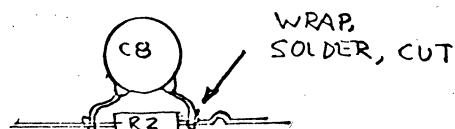
- 1) R2, 330 ohms, 1/4 watt, color code orange-orange-brown
- 2) R14, 100 ohms, 1/4 watt, color code brown-black-brown
- 3) D1, 1N5231B
- 4) C8, 0.047 uF disc ceramic

Assemble these parts as follows:

1. Form one lead of R2, R14, and the cathode (banded) lead of D1 for upright P.C. insertion as shown:

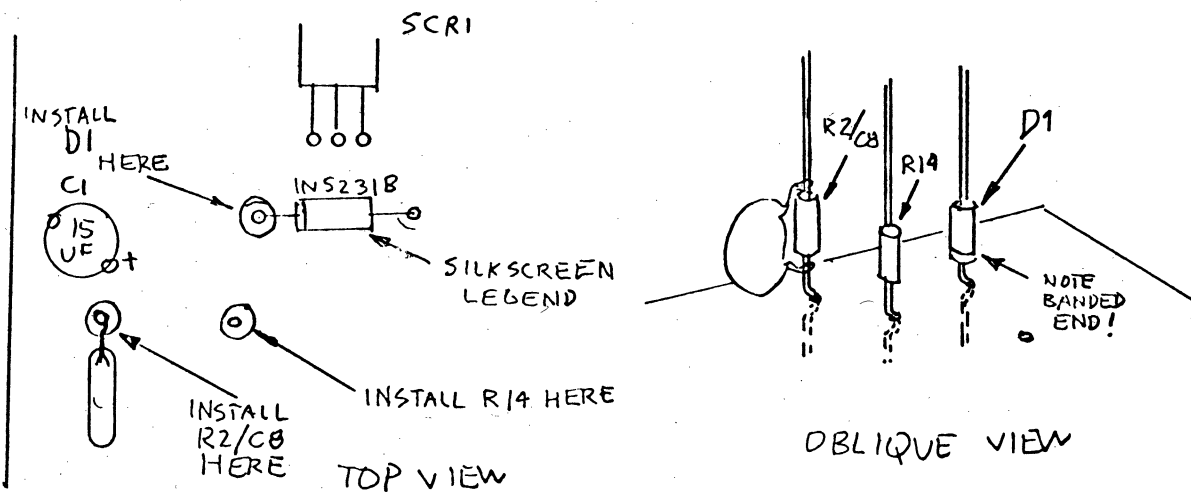


2. Solder C8 in parallel with R2 as shown:

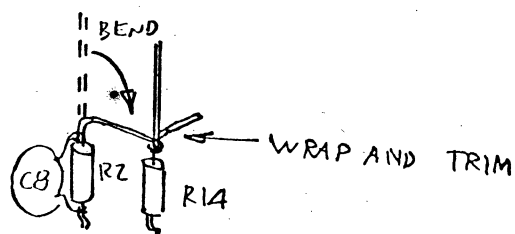


3. Install and solder R2-C8, R14, and D1 as shown below. Install the formed leads into the board with the unformed leads vertical. Position R2-C8 so that the body of C8 is parallel to the board edge and oriented away from C1.

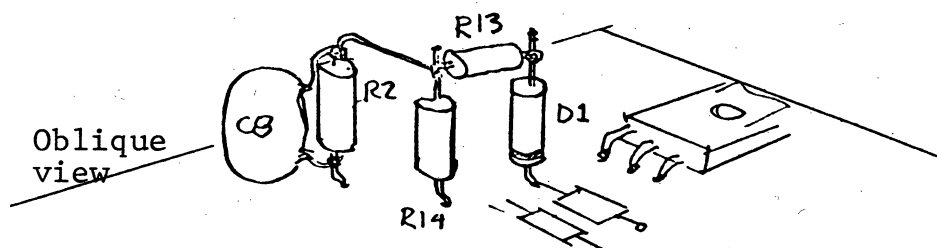




4. Bend the top lead of R2-C8 over towards R14, and bend it around the top lead of R14 one-eighth inch from the body of R14. Solder, and trim the excess lead of R2-C8 only.



5. Install R13 between the top leads of D1 and R14. Wrap R13's leads around R14 and D1 leads. Solder all connections at both points, and trim excess lead lengths. The resulting final configuration is shown below.



Schematic Diagram X-12 of the regulator includes these changes.



- ( ) Position Q1 (T1P41), with component nomenclature up, on heat sink so hole in Q1 package is aligned with the holes in sink and PC board. Observe how the leads of Q1 must be bent down to fit the pads for Q1 and bend them accordingly. Apply heat sink compound to Q1, the heat sink, and both sides of the rectangular mica insulator. Place mica insulator between heat sink and Q1, insert leads (emitter lead to right) and fasten Q1, insulator and heat sink to board with a 6-32 x  $\frac{1}{2}$  Nylon screw, lockwasher and nut. Insert screw from back (solder) side of board and drive nut finger tight.
- ( ) Position FWB1 (MDA970-1), with "+" lead to the right, on heat sink, determine how leads must be bent as you did for Q1, and bend leads. Apply heat sink compound. Insert leads ("+" lead to right) and fasten FWB1 and heat sink to PC board with a 4-40 x  $\frac{5}{8}$  screw, lockwasher and nut. Insert screw from back (solder) side of board and drive nut finger tight.
- ( ) Position SCR1 (IR106B2 or MCRL06-2) on heat sink with component nomenclature up and prepare it for installation as you did Q1 and FWB1. Apply heat sink compound to SCR1, the heat sink, and both sides of the circular mica insulator. Place the mica insulator between the heat sink and SCR1, insert leads and fasten SCR1, insulator and heat sink to PC board with a 4-40 x  $\frac{7}{16}$ " screw, lockwasher and nut. Insert screw from back (solder) side of board and drive nut finger tight.
- ( ) Check alignment of heat sink, SCR1, Q1 and FWB2 and tighten the three mounting screws. Solder all leads and trim if required. Wipe off excess heat sink compound, if necessary. NOTE: The heat sink may have to be repositioned when you mount the Sol-REG on the power supply subchassis. This will require that you loosen the mounting screws for SCR1, Q1 and FWB2 and retighten them after repositioning the heat sink.
- ( ) Step 21. Connect two wire cable assembly (C8 to Regulator Board cable) to regulator. Tin ends without lugs and solder green (+) lead to pad X2 and white (-) lead to pad X3.
- ( ) Step 22. Test Sol-REG for short circuits. Check for continuity between FWB1 (MDA970-1) mounting screw and the following points: (The resistance should be greater than 20 ohms in all cases.)
- |             |                     |                  |
|-------------|---------------------|------------------|
| X2          | Q1, Base            | D3, top lead     |
| T2          | Q1, Collector       | D4, top lead     |
| T1          | D1, right-hand lead | *D3, bottom lead |
| Q1, Emitter | R1, left-hand lead  | *D4, bottom lead |

\*Resistance will be initially low due to C4 and C5, but it should increase to greater than 20 ohms after a few seconds.



( / ) Step 23. Set Sol-REG to one side.

### 2.7.3 Power Supply Subchassis Assembly and Test

( / ) Step 24. Mount transformer (T1 for Sol-10, T2 for Sol-20) on power supply subchassis (L-shaped chassis).

Position transformer as shown in drawing on Page X-2 and attach it to the subchassis with three 8-32 x  $\frac{1}{2}$  binder or pan head screws, #8 lockwashers and 8-32 hex nuts. Insert screws from bottom and outer side of chassis as shown. Place lockwasher on each screw and secure loosely with hex nuts. Slide transformer as close as possible to the edge of the chassis and tighten nuts.

#### NOTE

Only one of the holes in the side wall is used. Use the one that lines up with the transformer mounting tab.

( / ) Step 25. Prepare transformer leads.

( / ) Twist the two black wires together except for the last two inches at the commoning block lug end.

( / ) Twist the two green wires together for their full length.

( / ) Twist the two yellow wires together for their full length.

\* ( / ) Twist the two blue wires together for their full length.

( / ) Step 26. Connect Sol-PC power cable (4-wire cable which connects to J10 on Sol-PC) to Sol-REG. Tin ends of cable and solder green lead to pad X9, white lead to pad X1, red lead to pad X7 and white-yellow lead to pad X8.

\* ( / ) Step 27. Connect Sol-20 DC power cable (5 wire) to Sol-REG. Tin ends of cable and solder white lead to pad ~~X4~~ (above R8), red-white lead to pad X5 (between C5 and FWB2) and yellow-white lead to pad X6 (left of C5).

( / ) Step 28. Connect transformer leads to Sol-REG.

( / ) Solder green leads to pads T1 and T2, white-yellow lead to pad T3 and yellow leads to pads T4 and T5 on Sol-REG circuit board.

( / ) Step 29. Prethread the three Sol-REG heat sink mounting holes in the power supply subchassis shown in drawing on page X-2 with #6 x 5/16 sheet metal screws. Remove screws.

# Processor Technology

Processor Technology  
Corporation

7100 Johnson Industrial Drive  
Pleasanton, CA 94566

(415) 829-2600  
Cable Address - PROCTEC

## Sol MANUAL

### CHANGE NOTICE #14

Step 27, on page II-14 calls for the connection of the white wire from the Sol-20 DC power cable to pad X4, a ground connection. This pad is too small; the wire should be run instead to the pad immediately to the right of pad T3, also a ground connection. This pad is shown on the legend as pad X-10. (NOTE: Some legends incorrectly show this pad as a second pad X-5. Make sure that the red-white lead goes to the pad labelled X-5 which is between C5 and FWB2.)

To avoid confusion when you reach this point in the assembly procedures, cross out "X4 (above R8)" in Step 27, and write in its place "X10 (to right of T3)". If pad X10 is incorrectly labelled X5, the note should read "X5 (to right of T3)". Also make a note reading "See Change Notice #14."



## Sol POWER SUPPLY

## SECTION II

- ( ) Step 30. Place #4 lockwashers on two 4-40 x 3/16 binder or pan head screws. Insert these screws from the bottom side of the power supply subchassis through the two mounting holes located near the middle of the bottom of the power supply subchassis, one on each side. Place another #4 lockwasher on the screws and drive each screw tightly into a 4-40 x 1/4 tapped spacer.
- ( ) Step 31. Position Sol-REG PC board with top edge over the previously installed spacers. Place #4 lockwashers on two 4-40 x 3/16 binder or pan head screws and drive screws through Sol-REG board into spacers.
- ( ) Step 32. Attach heat sink on Sol-REG to power supply subchassis as shown in drawing on Page X-2. At this point use only the two side screws which you used in Step 29 to pre-thread the holes. (The middle screw will be installed later.) Place a #6 lockwasher on each screw before driving it through the sink into the subchassis. Figure 2-4 shows an assembled Sol-10 power supply subchassis.

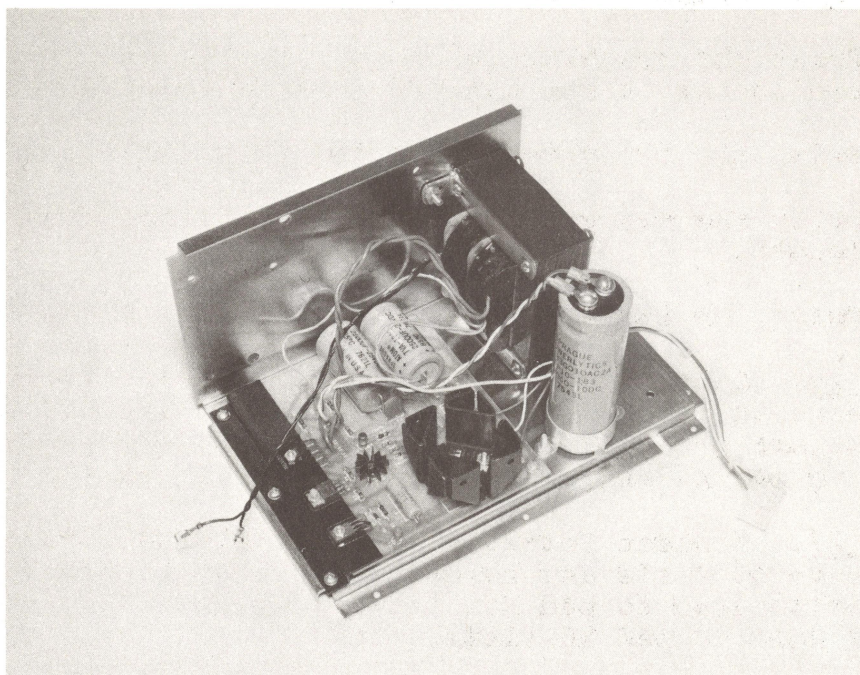


Figure 2-4. Sol-10 power supply subchassis assembly.  
(Rear of subchassis at left.)

- \* ( ) Step 33. Install bridge rectifier FWB3 on power supply subchassis.

(Step 33 continued on Page II-16.)



Position FWB3 (MDA980-1) on power supply subchassis as shown in drawing on Page X-2. BE SURE NEGATIVE (-) TERMINAL OF FWB3 is next to transformer. Insert a 6-32 x  $\frac{1}{2}$  binder or pan head screw from bottom of subchassis, place #6 lockwasher on screw and secure with 6-32 hex nut.

\* ( ) Step 34. Connect blue transformer wires to unmarked terminals of FWB3.

\* ( ) Step 35. Install large ( $2\frac{1}{2}$ " ) mounting ring for C9 (54,000 ufd capacitor) on side wall of power supply subchassis as shown in drawing on Page X-2.

Position ring over the three mounting holes in the side wall of subchassis so the clamping screw faces the bottom of subchassis and so it will be accessible from the Sol-REG end of the subchassis. Insert three 6-32 x  $\frac{1}{2}$  binder or pan head screws from outer side of side wall through the mounting holes. Place #6 lockwasher on each screw and secure with 6-32 hex nut. Figure 2-5 shows an assembled Sol-20 power supply subchassis.

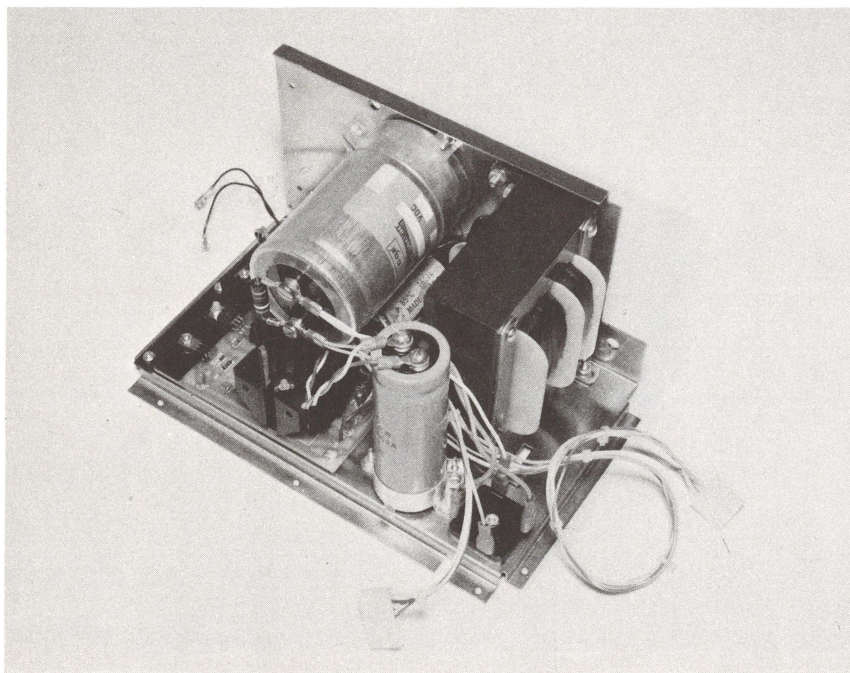


Figure 2-5. Sol-20 power supply subchassis assembly.  
(Rear of subchassis at left.)

( ) Step 36. Install small ( $1\frac{1}{2}$ " ) mounting ring for C8 (18,000 ufd capacitor) as shown in drawing on Page X-2.

(Step 36 continued on Page II-17.)

Position ring over the two mounting holes located between FWB3 and the Sol-REG so that the clamping screw is positioned between the transformer and FWB3. Insert two 6-32 x  $\frac{1}{2}$  binder or pan head screws from bottom side of chassis through the mounting holes. Place #6 lockwasher on each screw and secure with 6-32 hex nut. (Refer to Figure 2-4.)

- ( / ) Step 37. Route Sol-PC power cable between C8 mounting ring and the transformer, mount C8 in its mounting ring, and tighten clamping screw. (See Figure 2-4.)
- ( / ) Step 38. Connect white wire of C8 cable to negative (-) terminal of C8 and green wire to positive (+) terminal of C8. (This cable was soldered to the Sol-REG when you assembled it.) Remove terminal screws, place #10 lockwasher on each screw, place cable lugs on screws and drive screws tightly into appropriate terminals.
- \* ( ) Step 39. Mount C9 in its mounting ring with its "+" terminal slightly toward C8 and tighten clamping screw. (See Figure 2-5.)
- \* ( ) Step 40. Prepare R13 (39 ohm 2 watt) for installation on C9.  
Solder a #10 lug to each lead of R13. Bend leads of R13 to fit the terminals of C9. (R13 should fit on C9 as shown in Figure 2-5.)
- \* ( ) Step 41. Connect Sol-20 DC power cable (5 wire) and R13 to C9. Route cable between C8 and transformer.

Remove terminal screws from C9. Place lockwasher, terminal screw, blue lead of Sol-20 DC cable and one R13 lead on one terminal screw and drive it into the positive (+) terminal on C9. Attach lockwasher, white cable lead and other R13 lead to negative (-) terminal on C9 in the same manner. Tighten both capacitor terminals tightly.

#### CAUTION

LOOSE CONNECTIONS ON C9 CAN LEAD TO ARC-  
ING AND SUBSEQUENT POWER SUPPLY DAMAGE.

- \* ( / ) Step 42. Connect blue pigtail of Sol-20 DC cable to positive (+) terminal of FWB3. (This pigtail has a spade lug at its free end and is connected to the lug you just attached to the positive terminal of C9.) Connect white pigtail of Sol-20 DC cable to negative (-) terminal of FWB3. (This pigtail has a spade lug at its free end and is connected to the lug you just attached to the negative terminal of C9.)

- ( ) Step 43. Connect green lead from AC receptacle (mounted on fan closure plate) to power supply subchassis assembly as shown in drawing on Page X-2. (Use the #6 x  $\frac{1}{4}$  sheet metal screw with which you prethreaded the middle Sol-REG heat sink mounting hole in Step 29.) Place lug on screw and drive screw into the middle Sol-REG heat sink mounting hole.
- ( ) Step 44. Route black transformer leads along side wall of power supply subchassis out toward the Sol-REG heat sink. (See Figure 2-4.) Attach one lead to pin 2 of the commoning block (mounted on fan closure plate) nearest the fan. Attach other lead to pin 3 of the other commoning block.
- ( ) Step 45. Install cable tie wraps.
  - ( ) Install one wrap around the wires that connect to Sol-REG pads T1,2,3,X2 and X3 as shown in the Detail A - Wiring portion of the drawing on Page X-2.
  - \* ( ) Install another wrap around the leads from C9 as shown in Detail B of drawing on Page X-2.

Two other wraps are supplied with your kit. Use them as appropriate to make your power supply cabling neater.
- ( ) Step 46. Using a #6 x  $\frac{1}{4}$  sheet metal screw, attach fan closure plate to power supply subchassis as shown in Drawing No. X-2.
- ( ) Step 47. Push on-off switch in and out to determine the OFF position (switch mechanically out). With switch in OFF position, connect AC power cord to AC receptacle. Then plug power cord into 110 V ac outlet.
- ( ) Step 48. Test power supply for proper operation.
  - ( ) Make sure on-off switch is in OFF position.
  - ( ) Install fuse in fuse holder. CAUTION: NEVER INSTALL OR REMOVE FUSE WITH POWER ON.
  - ( ) Check connector on Sol-PC power cable (4 wire) to insure it is wired as shown in Figure 2-6.
  - \* ( ) Check connector on Sol-20 power cable (5 wire) to insure it is wired as shown in Figure 2-7.
  - ( ) Turn on-off switch ON.
  - ( ) Measure the voltages at the Sol-PC connector at the points indicated in Figure 2-6. The voltages must be as given in Figure 2-6. NOTE: Do not take voltage measurements at any other points in the power supply, even though they may be more accessible. It is important that the indicator voltages be available at the connector.

\*( ) Measure the voltages at the Sol-20 connector at the points indicated in Figure 2-7. The voltages must be within the ranges given in Figure 2-7. (See preceding NOTE.)

( ) If the power supply fails any of the preceding tests, locate and correct the cause before proceeding.

If the power supply is operating correctly, turn on-off switch OFF, disconnect power cord, set power supply to one side and go on to Section III.

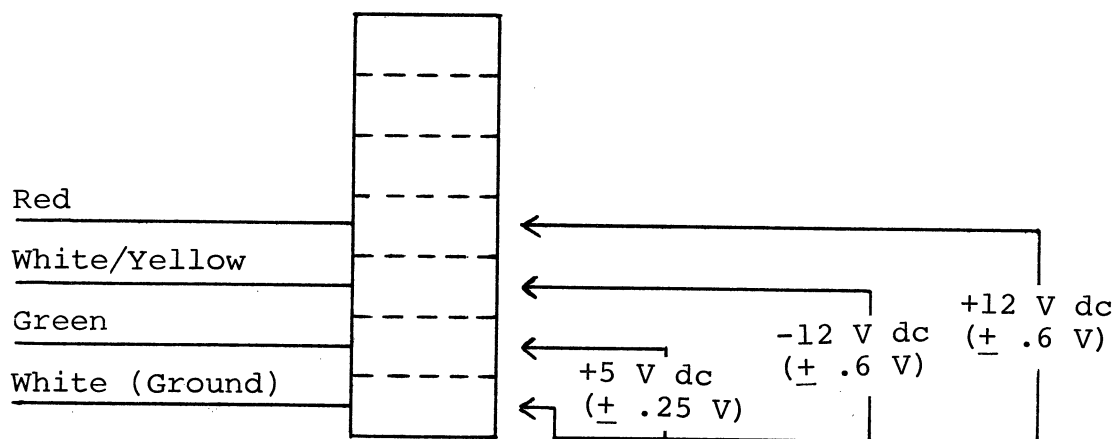


Figure 2-6. Sol-PC power connector and voltage measurements.

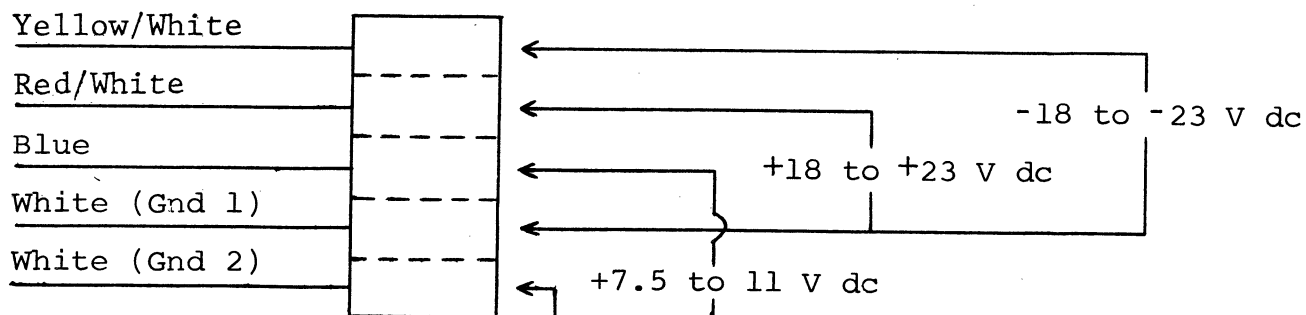


Figure 2-7. Sol-20 power connector and voltage measurements.





### III Sol-PC ASSEMBLY and TEST

3.1	Parts and Components . . . . .	III-1
3.2	Assembly Tips . . . . .	III-1
3.3	Assembly Precautions . . . . .	III-6
3.3.1	Handling MOS Integrated Circuits . .	III-6
3.3.2	Soldering . . . . .	III-6
3.3.3	Power Connection (J10) . . . . .	III-6
3.3.4	Installing and Removing Integrated Circuits . . . . .	III-6
3.3.5	Installing and Removing Personality Module . . . . .	III-6
3.3.6	Use of Clip Leads . . . . .	III-7
3.4	Required Tools, Equipment and Materials . .	III-7
3.5	Orientation (Sol-PCB) . . . . .	III-7
3.6	Sol-PC Assembly-Test Procedure . . . . .	III-7
3.6.1	Circuit Board Check . . . . .	III-8
3.6.2	Personality Module Assembly . . . .	III-9
3.6.3	Sol-PCB Assembly and Test . . . . .	III-9
3.6.4	Modification for 625 Line Video . . .	III-40





### 3.1 PARTS AND COMPONENTS

Check all parts and components against the "Parts List" on Pages III-2 through III-4 (Table 3-1). If you have difficulty in identifying any parts by sight, refer to Figure 3-1 on Page III-5.

### 3.2 ASSEMBLY TIPS

1. Scan Sections III and IV in their entirety before you start to assemble your Sol-PC kit.
2. In assembling your Sol-PC, you will be following an integrated assembly-test procedure. Such a procedure is designed to progressively insure that individual circuits and sections in the Sol-PC are operating correctly. IT IS IMPORTANT THAT YOU FOLLOW THE STEP-BY-STEP INSTRUCTIONS IN THE ORDER GIVEN.
3. Assembly steps and component installations are preceded by a set of parentheses. Check off each installation and step as you complete them. This will minimize the chances of omitting a step or component.
4. When installing components, make use of the assembly aids that are incorporated on the circuit boards and the assembly drawings. (These aids are designed to assist you in correctly installing the components.)
  - a. The circuit reference (R3, C10 and U20, for example) for each component is silk screened on the PC boards near the location of its installation.
  - b. Both the circuit reference and value or nomenclature (1.5K and 74H00, for example) for each component are included on the assembly drawings near the location of its installation.
5. To simplify reading resistor values after installation, install resistors so that the color codes or imprints read from left to right and top to bottom as appropriate (boards oriented as defined in Paragraph 3.5 on Page III-7).
6. Unless specified otherwise, install components, especially disc capacitors, as close as possible to the boards.
7. Should you encounter any problem during assembly, call on us for help if needed.

Table 3-1. Sol-PC Parts List.

<u>INTEGRATED CIRCUITS</u>	
1 AM0026 or DM0026 (U104)	1 74S04 (U92)
1 4N26 (U39)	2 7406 (U57,87)
1 8T94 (U58)	2 74LS10 (U47,61)
5 8T97 (U67,68,77,80,81)	3 74LS20 (U23,59,83)
2 1458CP or 1558CP (U56,108)	1 74LS86 (U74)
1 1489A (U38)	8 74LS109 (U43,52,63,64,70, 72,73,75)
2 TMS6011NC (U51,69)	1 74LS136 (U22)
1 MCM6574 or MCM6575 (U25)	3 74LS138 (U34,35,36)
1 4001 (U102)	3 74LS157 (U12,30,32)
2 4013 (U100,113)	4 74LS163 or 25LS163 (U28,31,33,40)
1 4019 (U111)	1 74166 (U41)
1 4023 (U98)	2 74173 (U95,96) & T10
1 4024 (U86)	1 74175 (U97)
1 4027 (U101)	9 74LS175 or 25LS175 (U2,13,26,27,42,76,90,93,106)
3 4029 (U1,11,84)	4 74LS253 (U65,66,78,79)
1 4030 (U99)	7 74LS367 (U29,37,50,71,89, 94,107)
2 4046 (U85,110)	1 8080, 8080A or 9080A (U105)
2 4049 (U88,109)	1 8836 or 8T380 (U46)
1 4520 (U112)	16 91L02APC or 2102L1PC (U3 - 10, U14 - 21)
1 74H00 (U91)	1 93L16 (U62)
3 74LS00 (U44,48,55)	
2 74LS02 or 9LS02 (U53,60)	
4 74LS04 (U24,45,49,54)	
<u>TRANSISTORS</u>	<u>DIODES</u>
2 2N2222 (Q4 & Q5)	9 1N4148 or 1N914 (D1,D3 - 10)
2 2N2907 or 2N3460 (Q1 & Q2)	1 1N5231B Zener Diode (D11)
1 2N4360 (Q3)	4 1N4001 (D2,12,13,14)
<u>CRYSTAL</u>	<u>RELAYS</u>
1 14.318 MHz in HC-18/U Case (XTAL)	2 DIP Reed, Sigma 191-TE1A15S (K1 & K2)



## PROCESSOR TECHNOLOGY CORPORATION

Sol-PC SINGLE BOARD TERMINAL COMPUTER<sup>TM</sup> (Continued)

SECTION III

Table 3-1. Sol-PC Parts List (Continued).

<u>RESISTORS</u>			<u>CAPACITORS</u>		
2	6.8	ohm, $\frac{1}{2}$ watt, 5%	1	10	pfd, disc
2	47	ohm, $\frac{1}{4}$ watt, 5%	1	330	pfd, disc
1	75	ohm, $\frac{1}{4}$ watt, 5%	1	470	pfd, disc
1	100	ohm, $\frac{1}{4}$ watt, 5%	3	680	pfd, monolythic or disc ceramic (labeled 681 and usually blue)
3	100	ohm, $\frac{1}{2}$ watt, 5%			
1	200	ohm, $\frac{1}{4}$ watt, 5%	6	.001	ufd, disc
13	<sup>270</sup> 330	ohm, $\frac{1}{4}$ watt, 5%	2	.001	ufd, Mylar tubular
1	330	ohm, $\frac{1}{2}$ watt, 5%	2	.01	ufd, Mylar tubular
<del>32</del>	470	ohm, $\frac{1}{4}$ watt, 5%	37	.047	ufd, disc
2	470	ohm, $\frac{1}{2}$ watt, 5%	12	.1	ufd, disc
9	680	ohm, $\frac{1}{4}$ watt, 5%	1	.1	ufd, Mylar tubular
63	1.5K	ohm, $\frac{1}{4}$ watt, 5%	1	.68	ufd, monolythic ceramic
1	3.3K	ohm, $\frac{1}{4}$ watt, 5%	1	1	ufd, tantalum dipped (usually orange or red)
6	5.6K	ohm, $\frac{1}{4}$ watt, 5%	5	15	ufd, tantalum dipped (usually orange or red)
<del>32</del> 1	10	K ohm, $\frac{1}{4}$ watt, 5%	1	100	ufd, aluminum electrolytic
<del>12</del>	15	K ohm, $\frac{1}{4}$ watt, 5%			
2	39	K ohm, $\frac{1}{4}$ watt, 5%			
<del>12</del>	47	K ohm, $\frac{1}{4}$ watt, 5%			
<del>32</del>	50	K ohm, Potentiometer			
<del>43</del>	<sup>100</sup> 100	K ohm, $\frac{1}{4}$ watt, 5%			
2	150	K ohm, $\frac{1}{4}$ watt, 5%			
2	1	M ohm, $\frac{1}{4}$ watt, 5%			
1	2.2M	ohm, $\frac{1}{4}$ watt, 5%			
2	3.3M	ohm, $\frac{1}{4}$ watt, 5%			

<u>CONNECTORS</u>	
1	25-pin Female, AMP206584-2 (J1)
1	25-pin Male, AMP206604-1 (J2)
2	20-pin Header, 3M3492-2002 (J3 & J4)
1	30-pin Right Angle Edge Connector, VIKING 3KH15/1JKC15 (J5)
2	Miniature Phone Jack (J6 & J7)
2	Subminiature Phone Jack (J8 & J9)
1	7-pin Male Locking Molex Connector (J10)
1	100-pin Edge Connector, TI H322150-0306A (J11)
1	Molex-type DC Power Cable, mates with J10 (prefabricated)

Table 3-1. Sol-PC Parts List (Continued).

MISCELLANEOUS

1	Sol-PCB Circuit Board	length of #24 bare wire
2	8-pin DIP Socket	
29	14-pin DIP Socket	
74	16-pin DIP Socket	
1	24-pin DIP Socket	
3	40-pin DIP Socket	
16	Augat Pins on Carrier	
2	DIP Switch, 6 position (S1 & S4)	
2	DIP Switch, 8 position (S2 & S3)	
1	4-foot Length 72-ohm Coaxial Cable	
1	Tie Wrap for Coaxial Cable	
2	Mounting Bracket, Sol-1040	
2	Card Guide, SAE1250F	
10	#4 Lockwasher, internal tooth	
2	#4 Insulating Washer	
4	4-40 x $\frac{1}{4}$ Binder Head Screw	
6	4-40 x $\frac{7}{16}$ Binder Head Screw	
2	4-40 x $\frac{5}{8}$ Binder Head Screw	
10	4-40 Hex Nut	
1	Length Solder	
1	Manual	
1	Personality Module Kit (See Section IV for contents.)	

PROCESSOR TECHNOLOGY CORPORATION

SOL-PC SINGLE BOARD TERMINAL COMPUTER™

SECTION III

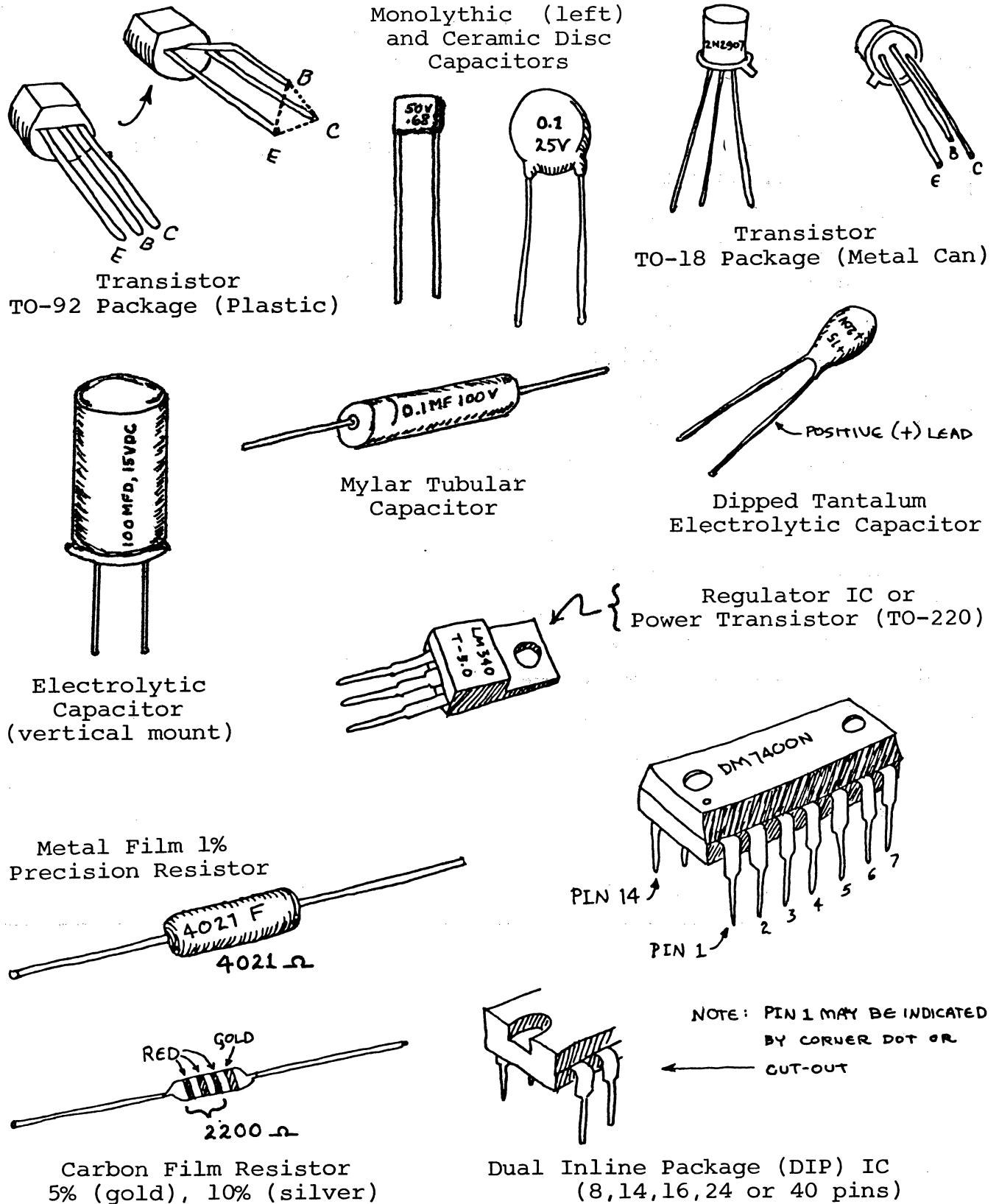


Figure 3-1. Identification of components.



### 3.3 ASSEMBLY PRECAUTIONS

#### 3.3.1 Handling MOS Integrated Circuits

Many of the IC's used in the Sol-PC are MOS devices. They can be damaged by static electricity discharge. Always handle MOS IC's so that no discharge will flow through the IC. Also, avoid unnecessary handling and wear cotton--rather than synthetic--clothing when you do handle these IC's.

#### 3.3.2 Soldering **\*\*IMPORTANT\*\***

1. Use a fine tip, low-wattage iron, 25 watts maximum.
2. DO NOT use excessive amounts of solder. DO solder neatly and as quickly as possible.
3. Use only 60-40 rosin-core solder. NEVER use acid-core solder or externally applied fluxes.
4. To prevent solder bridges, position iron tip so that it does not touch adjacent pins and/or traces simultaneously.
5. DO NOT press tip of iron on pad or trace. To do so can cause the pad or trace to "lift" off the board and permanently damage the board.
6. The Sol-PC uses circuit boards with plated-through holes. Solder flow through to the component (front) side of the board can produce solder bridges. Check for such bridges after you install each component.
7. The Sol-PC circuit boards have integral solder masks (a lacquer coating) that shield selected areas on the boards. This mask minimizes the chances of creating solder bridges during assembly. DO, however, check all solder joints for possible bridges.
8. Additional pointers on soldering are provided in Appendix IV of this manual.

#### 3.3.3 Power Connection (J10)

NEVER connect the DC power cable to the Sol-PC when power supply is energized. To do so can damage the Sol-PC.

#### 3.3.4 Installing and Removing Integrated Circuits

NEVER install or remove integrated circuits when power is applied to the Sol-PC. To do so can damage the IC.

#### 3.3.5 Installing and Removing Personality Module

NEVER install or remove the plug-in personality module when power is applied to the Sol-PC. To do so can damage the module.

### 3.3.6 Use of Clip Leads

TAKE CARE when using a clip lead to establish a ground connection when testing the Sol-PCB circuit board. Make sure that the clip makes contact only with the ground bus on the perimeter of the board.

### 3.4 REQUIRED TOOLS, EQUIPMENT AND MATERIALS

The following tools, equipment and materials are recommended for assembling and testing the Sol-PC:

1. Needle nose pliers
2. Diagonal cutters
3. Screwdriver
4. Sharp knife
5. Controlled heat soldering iron, 25 watt
6. 60-40 rosin-core solder (supplied)
7. Small amount of #24 solid wire
8. Volt-ohm meter
9. Video monitor or monochrome TV converted for video input.
10. IC test clip (optional)
11. Oscilloscope (optional)

### 3.5 ORIENTATION (Sol-PCB)

Location J5 (personality plug-in module connector) will be located in the upper right-hand area of the circuit board when location J10 (power connector) is positioned at the bottom of the board. In this position the component (front) side of the board is facing up and all IC legends (U1 through U10, U22 through U24, etc.) will read from left to right. Subsequent position references related to the Sol-PCB assume this orientation.

### 3.6 Sol-PC ASSEMBLY-TEST PROCEDURE

The Sol-PC is assembled and tested in sections and/or circuits. You will first test the Sol-PCB circuit board for shorts (solder bridges) between the power buses and ground. After assembling

the personality module (see Section IV), the clock and display control circuits are assembled. The bus, CPU, decoder and memory circuits are then assembled, followed by the parallel and serial input/output (I/O) and audio cassette I/O sections.

#### CAUTION

THE Sol-PC USES MANY MOS INTEGRATED CIRCUITS. THEY CAN BE DAMAGED BY STATIC ELECTRICITY DISCHARGE. HANDLE THESE IC'S SO THAT NO DISCHARGE FLOWS THROUGH THE IC. AVOID UNNECESSARY HANDLING AND WEAR COTTON, RATHER THAN SYNTHETIC, CLOTHING WHEN YOU DO HANDLE MOS IC'S. (STATIC CHARGE PROBLEMS ARE MUCH WORSE IN LOW HUMIDITY CONDITIONS.)

#### 3.6.1 Circuit Board Check

- ( ) Visually check Sol-PCB board for solder bridges (shorts) between traces, broken traces and similar defects.
- ( ) Check board to insure that the +5-volt-bus, +12 volt-bus and -12-volt bus are not shorted to each other or to ground. Using an ohmmeter, on "OHMS X 1K" or "OHMS X 10K" scale, make the following measurements (refer to Sol-PC Assembly Drawing X-3).
  - ( ) +5-volt Bus Test. Measure between positive and negative mounting pads for C58. There should be no continuity. (Meter reads close to "infinity" ohms.)
  - ( ) +12-volt Bus Test. Measure between positive and negative mounting pads for C59. There should be no continuity.
  - ( ) -12-volt Bus Test. Measure between positive and negative mounting pads for C60. There should be no continuity.
  - ( ) 5/12/(-12) Volt Bus Test. Measure between positive mounting pads for C58 and C59, between positive pad for C58 and negative pad for C60, and between positive pad for C59 and negative pad for C60. You should measure no continuity in any of these measurements.

If visual inspection reveals any defects, or you measure continuity in any of the preceding tests, return the board to Processor Technology for replacement. If the board is not defective, proceed to next paragraph.

## 3.6.2 Personality Module Assembly

Since the personality module is required for testing the Sol-PC in the later stages of its assembly, we suggest that you assemble the personality module first. In so doing, your Sol-PC assembly will proceed uninterrupted. Assembly instructions for the personality module are provided in Section IV of this manual.

If you wish to wait to assemble the personality module until it is needed, go on to Paragraph 3.6.3.

## 3.6.3 Sol-PCB Assembly and Test

Refer to Sol-PC assembly drawing X-3.

- ( ) Step 1. Install DIP sockets. Install each socket in the indicated location with its end notch oriented as shown on the circuit board and assembly drawing. Take care not to create solder bridges between the pins and/or traces. (Refer to footnotes at end of this step before installing U105.)

INSTALLATION TIP

Insert socket pins into mounting pads of appropriate location. On solder (back) side of board, bend pins at opposite corners of socket (e.g., pins 1 and 9 on a 16-pin socket) outward until they are at a 45° angle to the board surface. This secures the socket until it is soldered. Repeat this procedure with each socket until all are secured to the board. Then solder the unbent pins on all sockets. Now straighten the bent pins to their original position and solder.

<u>LOCATION</u>	<u>TYPE SOCKET</u>
(+) U1 through 21	16 pin
(+) U22 through 24	14 pin
(+) U25	24 pin
(+) U26 through 37	16 pin
(+) U38	14 pin
(+) U39	None
(+) U40 through 43	16 pin
(+) U44 through 49	14 pin
(+) U50	16 pin
(+) U51	40 pin
(+) U52	16 pin
(+) U53 through 55	14 pin
(+) U56	8 pin
(+) U57 through 61	14 pin

(Continued on Page III-10.)



## PROCESSOR TECHNOLOGY CORPORATION

Sol-PC SINGLE BOARD TERMINAL COMPUTER<sup>TM</sup>

SECTION III

<u>LOCATION</u>	<u>TYPE SOCKET</u>
( ) U62 through 68	16 pin
( ) U69	40 pin
( ) U70 through 73	16 pin
( ) U74	14 pin
( ) U75 through 81	16 pin
( ) U82#	None#
( ) U83	14 pin
( ) U84,85	16 pin
( ) U86,87	14 pin
( ) U88 through 90	16 pin
( ) U91,92	14 pin
( ) U93 through 97	16 pin
( ) U98 through 100	14 pin
( ) U101	16 pin
( ) U102	14 pin
( ) U103#	None #
( ) U104	None
( ) U105*	40 pin
( ) U106,107	16 pin
( ) U108	8 pin
( ) U109 through 112	16 pin
( ) U113	14 pin

#Spare locations, not used.

\*Note that U105 notch is positioned at the top.

- ( ) Step 2. Install the following capacitors in the indicated locations. Take care to observe the proper value, type and orientation, if applicable, for each installation. Bend leads outward on solder (back) side of board, solder and trim.

NOTE

Disc capacitor leads are usually coated with wax during the manufacturing process. After inserting leads through mounting holes, remove capacitor and clear the holes of any wax. Reinsert and install.

<u>LOCATION</u>	<u>VALUE (ufd)</u>	<u>TYPE</u>	<u>ORIENTATION</u>
( ) C1	.047	Disc	None
( ) C2	.047	"	"
( ) C3	.047	"	"
( ) C4	.047	"	"
( ) C5	.047	"	"
( ) C6	.047	"	"
( ) C7	.047	"	"
( ) C8	.047	"	"

<u>LOCATION</u>	<u>VALUE (ufd)</u>	<u>TYPE</u>	<u>ORIENTATION</u>
( ) C10	.047	Disc	None
( ) C11	.047	"	"
( ) C13	.047	"	"
( ) C14	.047	"	"
( ) C15	15	Tantalum	"+" lead bottom
( ) C16	.047	Disc	None

( ) Step 3. Check for +5-volt bus to ground shorts. Using an ohmmeter, measure between positive and negative mounting pads for C58. There should be no continuity. If there is, find and correct the problem before proceeding to Step 4.

( ) Step 4. Install the following capacitors in the indicated locations. Take care to observe the proper value, type and orientation, if applicable, for each installation. Bend leads outward on solder (back) side of board, solder and trim. (refer to NOTE in Step 2.)

<u>LOCATION</u>	<u>VALUE (ufd)</u>	<u>TYPE</u>	<u>ORIENTATION</u>
( ) C19	.047	Disc	None
( ) C20	.047	"	"
( ) C21	.047	"	"
( ) C24	.047	"	"
( ) C25	.047	"	"
( ) C26	.047	"	"
( ) C33	.047	"	"
( ) C38	.047	"	"
( ) C40	15	Tantalum	"+" lead bottom
( ) C41	.047	Disc	None
( ) C42	.047	"	"
( ) C45	.047	"	"
( ) C56	.047	"	"
( ) C58	15	Tantalum	"+" lead top
( ) C59	15	Tantalum	"+" lead top
( ) C60	15	Tantalum	"+" lead top
( ) C65	.047	Disc	None

( ) Step 5. Check for +5-volt bus to ground shorts. Using an ohmmeter, measure between the positive and negative leads of C58. You should measure at least 100 ohms. Less than 100 ohms indicates a short. If required, find and correct the problem before proceeding to Step 6. NOTE: In this and subsequent resistance measurements, any value greater than the minimum may normally occur, even much higher, unless otherwise indicated.

( ) Step 6. Install the following capacitors in the indicated locations. Take care to observe the proper value and type for each installation. Bend leads outward on solder (back) side of board, solder and trim. (Refer to NOTE in Step 2.)

(Step 6 continued on Page III-12.)

## PROCESSOR TECHNOLOGY CORPORATION

Sol-PC SINGLE BOARD TERMINAL COMPUTER<sup>TM</sup>

SECTION III

<u>LOCATION</u>	<u>VALUE (ufd)</u>	<u>TYPE</u>	<u>ORIENTATION</u>
( ) C9	.047	Disc	None
( ) C12	.047	"	"
( ) C17	.047	"	"
( ) C18	.047	"	"
( ) C22	.047	"	"
( ) C23	.047	"	"
( ) C27	.047	"	"
( ) C28	.047	"	"
( ) C46	.047	"	"

( ) Step 7. Check for +5-volt bus to ground shorts. Using an ohmmeter, measure between the positive and negative leads of C58. You should measure some resistance. Zero resistance indicates a short. If required, find and correct the problem before proceeding to Step 8.

( ) Step 8. Install diodes D8 (1N4148 or 1N914), D11 (1N5231B) and D12 (1N4001) in their locations (in the area below U90 through U92). Position D8 with its dark band (cathode) to the right, D11 with its band at the bottom, and D12 with its band at the top.

NOTE

The leads of D12 and its mounting holes are a snug fit. Take care when installing this diode.

( ) Step 9. Install the following resistors in the indicated locations. Bend leads to fit distance between mounting holes, insert leads, pull down snug to board, solder and trim.

<u>LOCATION</u>	<u>VALUE (ohms)</u>	<u>COLOR CODE</u>
( ) R104	10 K	brown-black-orange
( ) R105	1.5K	brown-green-red
( ) R106	1.5K	" " "
( ) R130	100, $\frac{1}{2}$ watt	brown-black-brown
( ) R131	100, $\frac{1}{2}$ watt	" " "
( ) R132	100, $\frac{1}{2}$ watt	" " "
( ) R133	330	orange-orange-brown
( ) R134	330	" " "
( ) R135 & 136	10 K	brown-black-orange
( ) R137 & 138	47	yellow-violet-black

( ) Step 10. Install the following capacitors in the indicated locations. Take care to observe the proper value and type for each installation. Bend leads outward on solder (back) side of board, solder and trim. (Refer to NOTE in Step 2.)

## PROCESSOR TECHNOLOGY CORPORATION

Sol-PC SINGLE BOARD TERMINAL COMPUTER<sup>TM</sup>

SECTION III

<u>LOCATION</u>	<u>VALUE</u>	<u>TYPE</u>
(✓) C39	.1 ufd	Disc
(✓) C43	680 pfd	Monolythic or Disc
(✓) C44	680 pfd	Monolythic or Disc
(✓) C61	.001 ufd	Disc
(✓) C62	.68 ufd	Monolythic
(✓) C63	.1 ufd	Disc
(✓) C64	10 pfd	Disc

- (✓) Step 11. Install 14.318 MHz crystal in its location just above C61. Insert leads and pull down until the case is 1/16" above the front surface of the board. Solder quickly and trim.
- (✓) Step 12. Install male Molex connector in location J10. Position connector so the locking clip is facing the crystal (XTAL), insert shorter pins in mounting holes and solder.
- (✓) Step 13. In the jumper area labeled CLK on the assembly drawing (between U90 and U91), install Augat pins in mounting holes A,B,C,D and E. (Refer to "Installing Augat Pins" in Appendix IV.) Using #24 bare wire, install a jumper between the A and B pins and another jumper between the D and E pins.
- ( ) Step 14. Install the following IC's in the indicated locations. Pay careful attention to the proper orientation. DO NOT SUBSTITUTE FOR ANY OF THESE IC's.

NOTE

Dots on the assembly drawing and PC board indicate the location of pin 1 of each IC.

<u>IC NO.</u>	<u>TYPE</u>
(✓) U77	8T97
(✓) U90	74LS175 or 25LS175
(✓) U91	74H00
(✓) U92	74S04
( ) U104*	AM0026 or DM0026*

\*Solder this IC in its location.  
See "Loading DIP Devices" in  
Appendix IV.

- ( ) Step 15. Connect power to power connector J10. Power and interconnection requirements are as follows:

(Step 15 continued on Page III-14.)

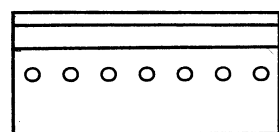


CAUTION 1

NEVER CONNECT POWER CABLE TO J10 WITH POWER SUPPLY ENERGIZED.

CAUTION 2

MAKE SURE POWER CABLE CONNECTOR MATES EXACTLY WITH J10; THAT IS, PIN 1 TO PIN 1, PIN 2 TO PIN 2, ETC. ANY OTHER MATING RELATIONSHIP WILL "BLOW" THE IC's.



1 2 3 4 5 6 7  
(J10, Top View)

<u>J10 PIN NO.</u>	<u>POWER</u>
1	Ground
2 and 6	+5 V dc $\pm 5\%$ , 2 A max
3 and 5	-12 V dc $\pm 5\%$ , 300 mA max
4	+12 V dc $\pm 5\%$ , 100 mA max
7	Ground

NOTE

Though not labeled on the connector, J10 pins are designated 1 through 7, reading from left to right.

- ( ) Step 16. Check clock circuits. If you have an oscilloscope, use part A of this step. If you do not, use part B.

A. Oscilloscope Check

- ( ) Using an oscilloscope, check for the waveforms given in Figure 3-2 on Page III-15 at the indicated observation points and in the order given. The waveforms shown in Figure 3-2 approximate actual waveforms. If any waveforms are incorrect, determine and correct the cause before proceeding with assembly.

NOTE

Irregularities up to 1 volt are acceptable on positive portions of waveforms. Negative portions, however, should be relatively flat.

B. Volt-ohm Meter Check

- ( ) Using the test probe shown in Figure 3-3 on Page III-16, set meter to DC volts and make the following measurements:

(Volt-ohm Meter Check continued on Page III-16.)

## PROCESSOR TECHNOLOGY CORPORATION

Sol-PC SINGLE BOARD TERMINAL COMPUTER<sup>TM</sup>

SECTION III

INVERTED

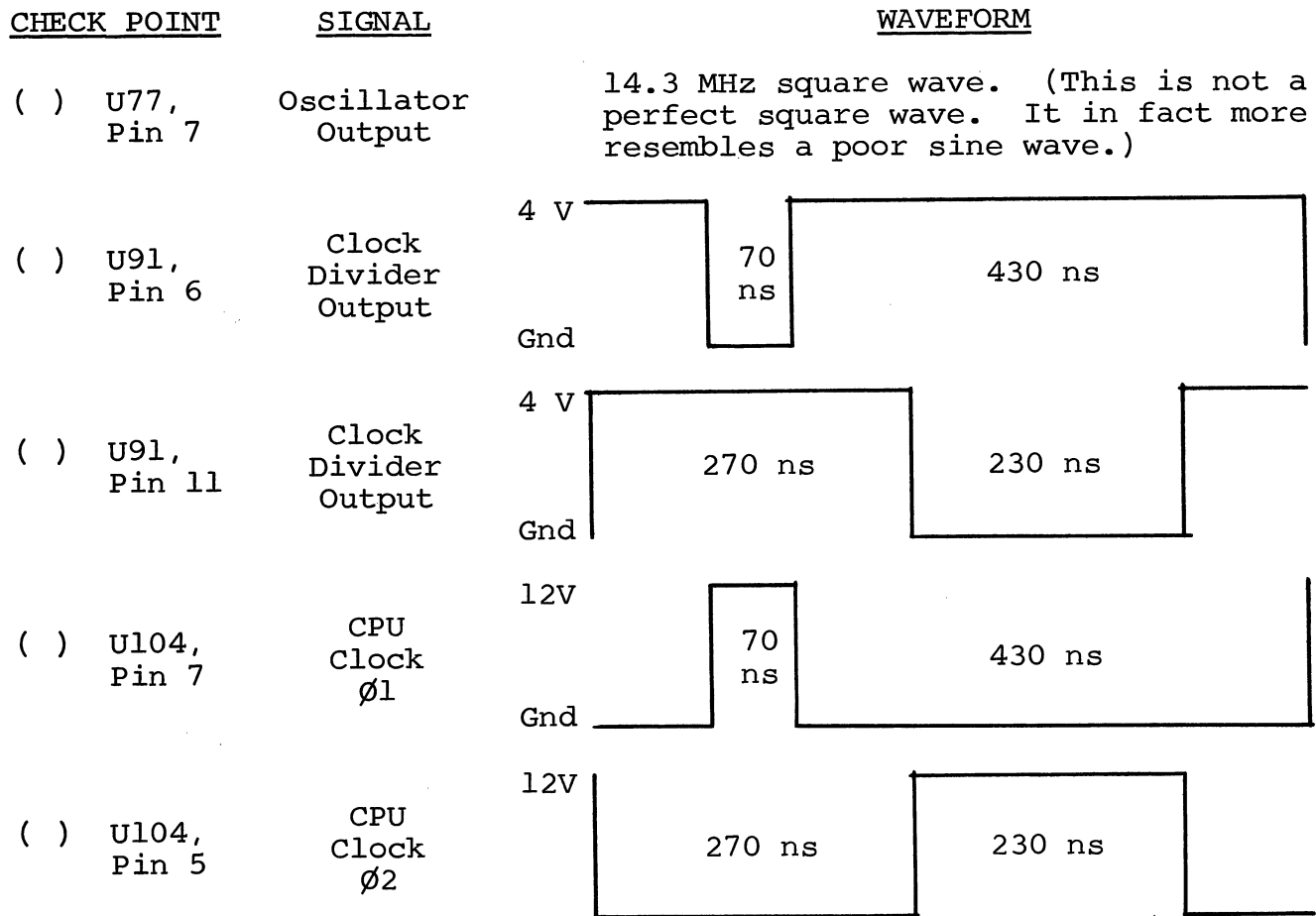


Figure 3-2. Clock circuit waveforms.

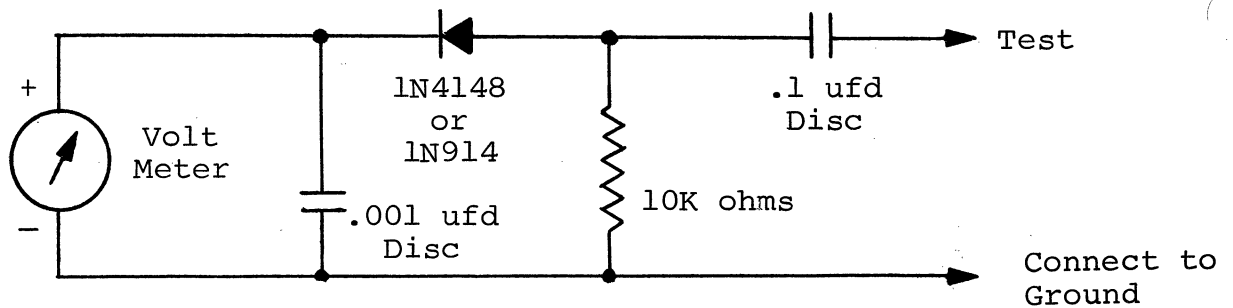


Figure 3-3. Test probe for Steps 16B and 25B.

NOTE 1

The probe shown in Figure 3-3 can be made using parts supplied with your Sol-PC kit. Since these parts will be used later in the Sol-PC assembly, DO NOT shorten the leads or otherwise alter the components. Assemble the probe using tack soldering technique.

NOTE 2

Make sure you have a good ground connection between the meter, probe and Sol-PCB.

- ( ) At pin 7 of U77 you should measure 1.5 V dc or higher. (A significantly lower reading indicates a faulty oscillator circuit.)
- ( ) At pin 6 of U91 you should measure 0.25 V dc or higher. (A significantly lower reading indicates a faulty clock divider, U90.)
- ( ) At pin 11 of U91 you should measure 1.25 V dc or higher. (A significantly lower reading indicates a faulty clock divider, U90.)
- ( ) At pin 5 of U104 you should measure 4 V dc or higher. (A significantly lower reading indicates a problem with U104.)
- ( ) At pin 7 of U104 you should measure 8 V dc or higher. (A significantly lower reading indicates a problem with U104.)
- ( ) If any voltages are incorrect, correct the problem before proceeding; if correct, turn off the power supply and disconnect the power cable.

## PROCESSOR TECHNOLOGY CORPORATION

Sol-PC SINGLE BOARD TERMINAL COMPUTER<sup>TM</sup>

SECTION III

- ( ) Step 17. Install the following resistors in the indicated locations. Bend leads to fit distance between mounting holes, insert leads, pull down snug to board, solder and trim.

<u>LOCATION</u>	<u>VALUE (ohms)</u>	<u>COLOR CODE</u>
( ) R1	1.5K	brown-green-red
( ) R2	1.5K	" " "
( ) R3	1.5K	" " "
( ) R4	1.5K	" " "
( ) R5	1.5K	" " "
( ) R6	1.5K	" " "
( ) R7	1.5K	" " "
( ) R8	1.5K	" " "
( ) R9	1.5K	" " "
( ) R10	1.5K	" " "
( ) R11	1.5K	" " "
( ) R16	1.5K	" " "
( ) R17	1.5K	" " "
( ) R19	1.5K	" " "
( ) R30	1.5K	" " "
( ) R80*	330, $\frac{1}{2}$ watt	orange-orange-brown
( ) R81	75	violet-green-black
( ) R82	200	red-black-brown
( ) R83	1.5K	brown-green-red
( ) R84	3.3M	orange-orange-green
( ) R85	1.5K	brown-green-red
( ) R86	1.5K	" " "
( ) R87	330	orange-orange-brown
( ) R88	680	blue-gray-brown
( ) R89	1.5K	brown-green-red
( ) R90	1.5K	" " "
( ) R96	1.5K	" " "
( ) R97	1.5K	" " "
( ) R98	10 K	brown-black-orange
( ) R99	1.5K	brown-green-red
( ) R100	10 K	brown-black-orange
( ) R101	1.5K	brown-green-red
( ) R102	3.3M	orange-orange-green
( ) R103	1.5K	brown-green-red
( ) R120	100 K	brown-black-yellow
( ) R121	10 K	brown-black-orange
( ) R122	10 K	" " "
( ) R123	39 K	orange-white-orange
( ) R124	1.5K	brown-green-red
( ) R125	1.5K	" " "
( ) R126	39 K	orange-white-orange
( ) R127	10 K	brown-black-orange
( ) R128	3.3K	orange-orange-red
( ) R129	10 K	brown-black-orange
( ) VR1 & VR2	50 K	Potentiometer

\*The leads of R80 and its mounting holes form a snug fit. Take care when installing this resistor.

## PROCESSOR TECHNOLOGY CORPORATION

Sol-PC SINGLE BOARD TERMINAL COMPUTER<sup>TM</sup>

SECTION III

- ( ) Step 18. Install the following capacitors in the indicated locations. Take care to observe the proper value and type for each installation. Bend leads outward on solder (back) side of board, solder and trim. (Refer to NOTE in Step 2.)

CAUTION

REFER TO FOOTNOTE AT END OF THIS STEP BEFORE  
INSTALLING C31.

<u>LOCATION</u>	<u>VALUE</u>	<u>TYPE</u>
( ) C31*	100 ufd	Aluminum Electrolytic
( ) C32	.1 ufd	Disc
( ) C34	680 pfd	Monolythic or Disc
( ) C35	.1 ufd	Mylar Tubular
( ) C36	.1 ufd	Disc
( ) C37	.1 ufd	Disc
( ) C52	.001 ufd	Mylar Tubular
( ) C53	.01 ufd	Mylar Tubular
( ) C54	.001 ufd	Disc
( ) C55	.001 ufd	Disc
( ) C57	.1 ufd	Disc

\*Install C31 with "+" lead at the top.

- ( ) Step 19. Install Q2 (2N2907 or 2N3460) in its location below and to the right of U88. The emitter lead (closest to tab on can) is oriented toward the left of the board and the base is oriented toward the bottom. Push straight down on transistor until it is stopped by the leads. Solder and trim.

- ( ) Step 20. Install diodes D9 and D10 (1N4148 or 1N914) in their locations below U88. Position D9 with its dark band (cathode) to the left and D10 with its band to the right.

- ( ) Step 21. Install coaxial cable, composite video output. (See Figure 3-4 for details on how to prepare cable.)

- ( ) Strip away about 1¼" of the outer insulation to expose shield. Unbraid shield, gather and twist into a single lead. Then strip away the inner conductor insulation, leaving about ¼" at the shield end.

CAUTION

WHEN PREPARING AND INSTALLING SHIELD, BE  
SURE BITS OF BRAID DO NOT FALL ONTO BOARD.  
SUCH DEBRIS CAN CREATE HARD-TO-FIND SHORT  
CIRCUITS.

- ( ) Insert inner conductor in mounting hole P1 (left side of board), solder and trim.



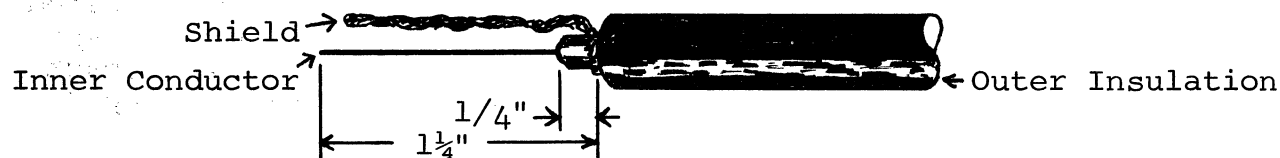


Figure 3-4. Coaxial cable preparation.

- (~~1~~) Insert twisted shield in mounting hole P2, solder and trim. Using the two large holes to the right of VR1 and VR2, tie cable to board with tie wrap (see CAUTION below).

CAUTION

AFTER INSTALLATION, FINE BITS OF THE BRAID FROM THE SHIELD MAY WORK LOOSE AND FALL ONTO THE BOARD AND CREATE HARD-TO-FIND SHORT CIRCUITS. TO PREVENT THIS, COAT ALL EXPOSED BRAID WITH AN ADHESIVE AFTER SOLDERING AND TIEING. USE AN ADHESIVE SUCH AS SILICONE, CONTACT CEMENT OR FINGERNAIL POLISH. DO NOT USE WATER BASE ADHESIVES.

- (~~2~~) Step 22. Install 6-position DIP switch in location S1 on left end of board. Position Switch No. 1 at the bottom.
- (~~3~~) Step 23. Install 20-pin header in location J4 (video expansion connector) between U28 and U29. Position header so pin 1 is in the lower right corner. (An arrow on the connector points to pin 1.)
- (~~4~~) Step 24. Install the following IC's in the indicated locations. Pay careful attention to the proper orientation.

NOTE

Dots on the assembly drawing and PC board indicate the location of pin 1 of each IC.

<u>IC NO.</u>	<u>TYPE</u>
( <del>5</del> ) U28	74LS163 or 25LS163
( <del>6</del> ) U31	74LS163 or 25LS163
( <del>7</del> ) U33	74LS163 or 25LS163
( <del>8</del> ) U40	74LS163 or 25LS163
( <del>9</del> ) U43	74LS109
( <del>10</del> ) U47	74LS10
( <del>11</del> ) U49	74LS04

(Step 24 continued on Page III-20.)

## PROCESSOR TECHNOLOGY CORPORATION

Sol-PC SINGLE BOARD TERMINAL COMPUTER<sup>TM</sup>

SECTION III

<u>IC NO.</u>	<u>TYPE</u>
<del>( )</del> U59	74LS20
<del>( )</del> U60	74LS02 or 9LS02
<del>( )</del> U62	93L16
<del>( )</del> U74	74LS86
<del>( )</del> U75	74LS109
<del>( )</del> U87	7406
<del>( )</del> U88*	4049*
( ) U102*	4001*

\*MOS device. Refer to CAUTION on Page III-8.

- ( ) Step 25. Apply power to Sol-PC and check display section timing chain operation. If you have an oscilloscope, use part A of this step. If you do not, use part B.

A. Oscilloscope Check

- ( ) Using an oscilloscope, check for the waveforms given in Figure 3-5 at the indicated observation points and in the order given. The waveforms shown in Figure 3-5 approximate actual waveforms. If any waveforms are incorrect, determine and correct the cause before proceeding with assembly.

NOTE

Irregularities up to 1 volt are acceptable on positive portions of waveforms. Negative portions, however, should be relatively flat.

B. Volt-ohm Meter Check

- ( ) Using the test probe made in Step 16B, measure the voltage at pin 12 of U28. You should measure approximately 1 V dc. If you get a significantly lower reading, find and correct the cause before you proceed with assembly.

- ( ) Turn off power supply and disconnect power connector.

- ( ) Step 26. Check synchronization circuits.

- ( ) Set all S1 switches to OFF.

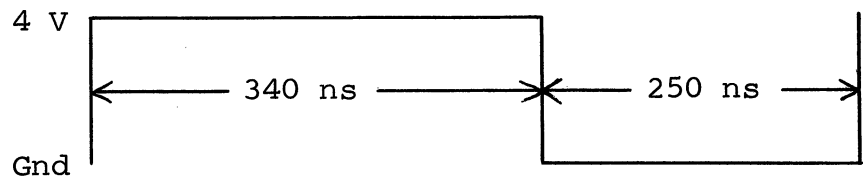
- ( ) Connect Sol-PC video output cable to video monitor.

SEE CAUTION ON PAGE III-22 BEFORE CONNECTING MONITOR.

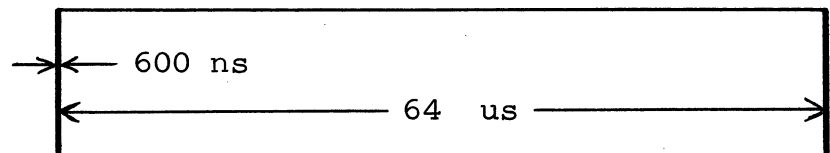
(Step 26 continued on Page III-22.)

CHECK POINTWAVEFORM

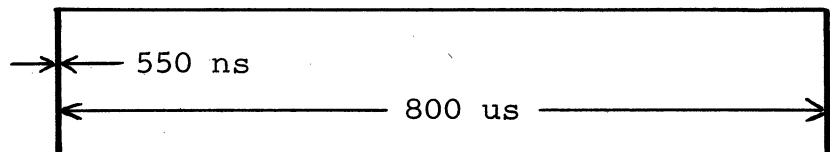
( ) U28, Pin 12



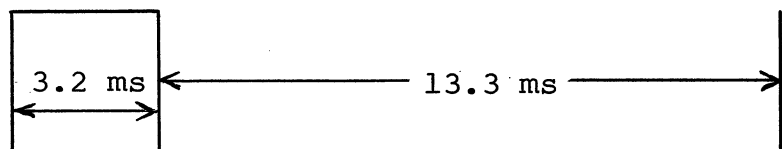
( ) U47, Pin 8



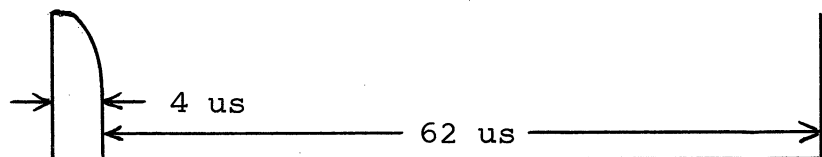
( ) U59, Pin 8



( ) U43, Pin 9



( ) U88, Pin 10



( ) U88, Pin 4

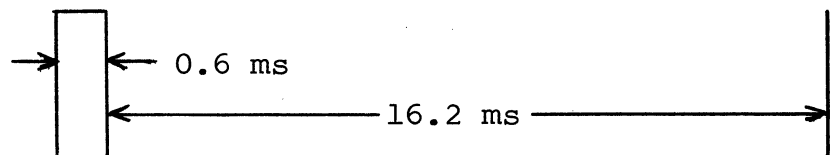


Figure 3-5. Display section timing waveforms.

CAUTION

DO NOT CONNECT THE Sol-PC VIDEO OUTPUT TO A MONITOR OR TV RECEIVER THAT IS NOT EQUIPPED WITH AN ISOLATION TRANSFORMER. (SEE PAGE AVI-7 IN APPENDIX VI.)

- ( ) Set VR2 (VERT) and VR1 (HORIZ) on the Sol-PC to their mid-range settings. Turn monitor on and apply power to the Sol-PC.

- ( ) The display raster will be pulled in. Using the monitor Vertical Hold, you should be able to obtain a slow roll (black horizontal bar moves slowly down the screen) and a stationary raster. Using the monitor Horizontal Hold, you should be able to adjust for an out of sync raster (numerous black lines cutting across the raster) and a stable raster. If you cannot obtain these conditions, locate and correct the cause before proceeding.

NOTE

For a stable presentation, a few monitors (especially modified TV sets) may require a higher sync amplitude than that supplied by the Sol-PC. In such cases, increase sync amplitude by reducing the value of R80. DO NOT DECREASE R80 BELOW 225 OHMS.

- ( ) If the synchronization circuits are operating correctly, turn monitor and power supply off, disconnect the power cable and go on to Step 27.
- ( ) Step 27. Install the following IC's in the indicated locations. Pay careful attention to the proper orientation.

NOTE

Dots on the assembly drawing and PC board indicate the location of pin 1 of each IC.

(Step 27 continued on Page III-23.)

<u>IC NO.</u>	<u>TYPE</u>
( ) U1*	4029*
( ) U2	74LS175 or 25LS175
( ) U11*	4029*
( ) U12	74LS157
( ) U13	74LS175 or 25LS175
( ) U25*	MCM6574 or MCM6575*
( ) U26	74LS175 or 25LS175
( ) U27	74LS175 or 25LS175
( ) U29	74LS367
( ) U30	74LS157
( ) U32	74LS157 or 25LS157
( ) U41	74166
( ) U42	74LS175 or 25LS175
( ) U44	74LS00
( ) U61	74LS10
( ) U89	74LS367

\*MOS device. Refer to CAUTION on Page III-8.

( ) Step 28. Check display circuits.

( ) Set S1 switches as follows:

No. 1 through 5: OFF

No. 6: ON

( ) Remove U42 and bend pin 6 out 45° to its normal position.  
(See Figure 3-6.) Re-install U42 with pin 6 out of the socket.

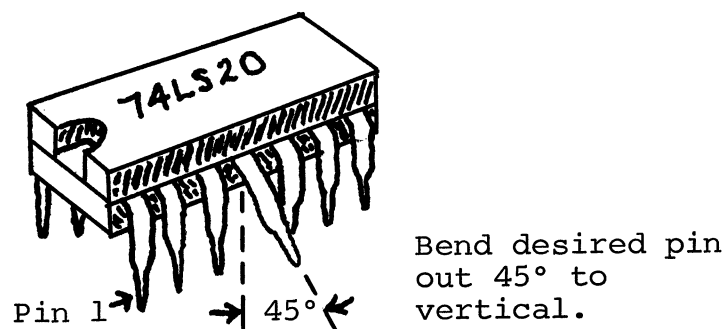


Figure 3-6. Bending selected pins on U42, 59 and 75 (U59 shown).

( ) Remove U59 and bend pin 4 in same manner as U42. Re-install U59 with pin 4 out of the socket.

(Step 28 continued on Page III-24.)



## PROCESSOR TECHNOLOGY CORPORATION

Sol-PC SINGLE BOARD TERMINAL COMPUTER<sup>TM</sup>

## SECTION III

- ( ) Remove U75 and bend pin 5 in same manner as U42. Re-install U75 with pin 5 out of the socket.
- ( ) Using #24 wire, install the following TEMPORARY jumpers in the sockets for U14 through U21. Double check jumpers after installing for correctness. (See Figure 3-7.)

<u>IC SOCKET</u>	<u>JUMPER</u>
( ) U14	Pin 12 to 6
( ) U15	Pin 12 to 5
( ) U16	Pin 12 to 4
( ) U17	Pin 12 to 8
( ) U18	Pin 12 to 2
( ) U19	Pin 12 to 7
( ) U20	Pin 12 to 1
( ) U21	Pin 12 to 16

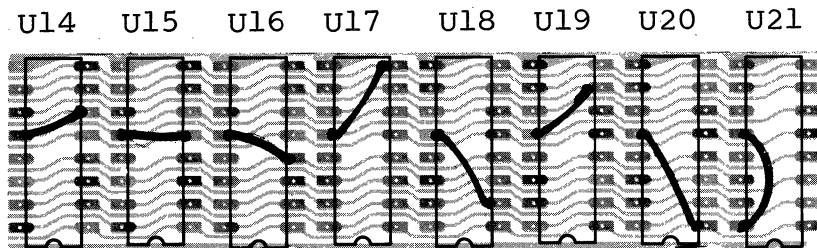


Figure 3-7. U14 through U21 socket jumpers.

- ( ) Turn monitor on and apply power to Sol-PC.
- ( ) Momentarily ground pin 1 of U2 and pin 5 of U75. The display shown in Figure 3-8 should appear on the monitor screen. *AR105 11/2/85*
- ( ) If the display circuits do not pass this test, determine and correct the cause before proceeding with assembly.
- ( ) If the display circuits are operating correctly:
  - ( ) Turn monitor and power supply off and disconnect the power cable.
  - ( ) Remove jumpers from U14 through U21 sockets.
  - ( ) Bend pin 6 on U42, pin 4 on U49 and pin 5 on U75 back to their normal position and re-install these three IC's in their appropriate sockets.

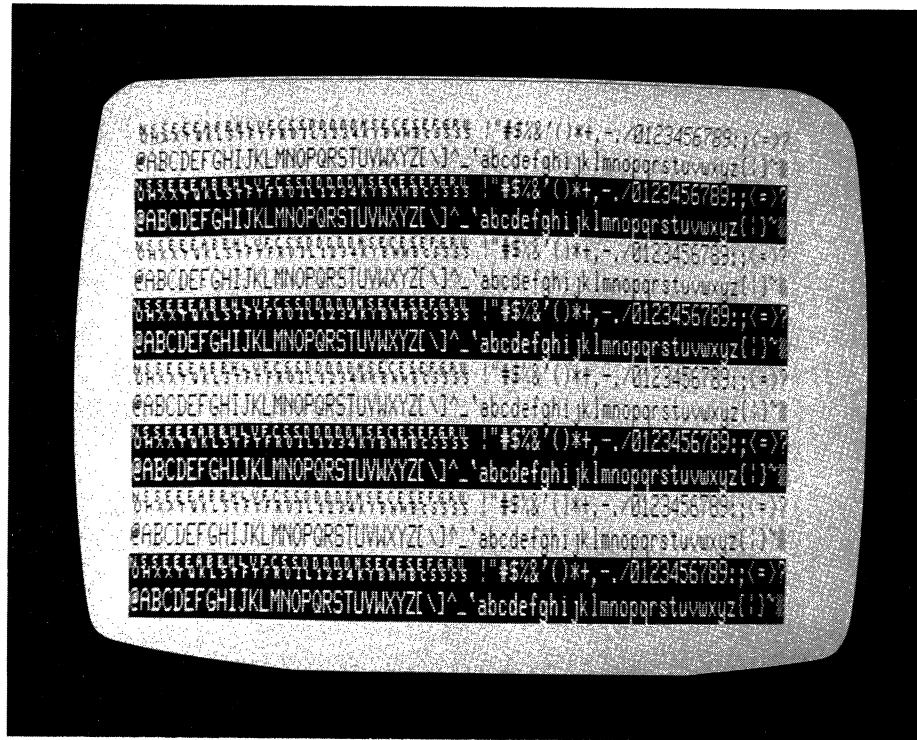


Figure 3-8. Display circuits test pattern with 6575 character generator as U25. 6574 is the same except graphic control characters are displayed.

*DO STEP 73 (P III-39) NOW*  
 ( ) Step 29. Install 91L02APC or 2102L1PC IC's in locations U14 through U21. Dots on the assembly drawing and PC board legend indicate the location of pin 1 of each IC.

#### CAUTION

IC's U14 THROUGH U21 ARE MOS DEVICES. REFER TO CAUTION ON PAGE III-8 BEFORE YOU INSTALL THESE IC's.

( ) Step 30. Install the following resistors in the indicated locations. Bend leads to fit distance between mounting holes, insert leads, pull down snug to board, solder and trim.

<u>LOCATION</u>	<u>VALUE (ohms)</u>	<u>COLOR CODE</u>
R12	1.5K	brown-green-red
R18	10 K	brown-black-orange

(Step 30 continued on Page III-26.)

<u>LOCATION</u>	<u>VALUE (ohms)</u>	<u>COLOR CODE</u>
(X) R20	1.5K	brown-green-red
(X) R31	1.5K	" " "
( ) R32	1.5K	" " "
( ) R33	1.5K	" " "
( ) R34	1.5K	" " "
(X) R35	1.5K	" " "
(X) R36	1.5K	" " "
( ) R41	1.5K	" " "
( ) R50	1.5K	" " "
( ) R51	1.5K	" " "
( ) R52	1.5K	" " "
( ) R53	1.5K	" " "
( ) R54	1.5K	" " "
( ) R55	1.5K	" " "
( ) R56	1.5K	" " "
( ) R57	1.5K	" " "
(X) R58	330	orange-orange-brown
( ) R107	10 K	brown-black-orange
( ) R108	10 K	" " "
( ) R109	10 K	" " "
( ) R110	10 K	" " "
( ) R111	10 K	" " "
( ) R112	10 K	" " "
( ) R113	10 K	" " "
( ) R114	10 K	" " "
( ) R115	1.5K	brown-green-red

(X) Step 31. Install diode D7 (1N4148 or 1N914) in its location between U46 and U47. Position D7 with its dark band (cathode) at the bottom.

( ) Step 32. Install 20-pin header in location J3 (keyboard interconnect) between U64 and U65. Position header so pin 1 is in the upper left corner. (An arrow on the connector points to pin 1.)

( ) Step 33. In the jumper area labeled PHTM on the assembly drawing (below U64), install Augat pins in mounting holes F and G. (Refer to "Installing Augat Pins" in Appendix IV.) Using #24 bare wire, install a jumper between pins F and G.

( ) Step 34. In the jumper area labeled RST on the assembly drawing (between U76 and U77), install Augat pins in mounting holes N and P. (Refer to "Installing Augat Pins" in Appendix IV.) Using #24 bare wire, install a jumper between pins N and P.

## PROCESSOR TECHNOLOGY CORPORATION

Sol-PC SINGLE BOARD TERMINAL COMPUTER<sup>TM</sup>

## SECTION III

- ( / ) Step 35. Install the following IC's in the indicated locations. Pay careful attention to the proper orientation.

NOTE

Dots on the assembly drawing and PC board indicate the location of pin 1 of each IC.

<u>IC NO.</u>	<u>TYPE</u>
( / ) <del>U105</del>	74LS175
( / ) <del>U45</del>	74LS04
( / ) <del>U46</del>	8T380 or 8836
( / ) <del>U48</del>	74LS00
( / ) <del>U50</del>	74LS367
( / ) <del>U54</del>	74LS04
( / ) <del>U63</del>	74LS109
( / ) <del>U64</del>	74LS109
( / ) <del>U67</del>	8T97
( / ) <del>U68</del>	8T97
( / ) <del>U76</del>	74LS175
( / ) <del>U94</del>	74LS367
( / ) <del>U107</del>	74LS367
( / ) <del>U93</del>	74LS175

- ( / ) Step 36. Apply power to Sol-PC and make the following voltage measurements:

<u>MEASUREMENT POINT</u>	<u>VOLTAGE*</u>
Pin 11 of U105 Socket	-5 V dc $\pm$ .25 V
Pin 20 of U105 Socket	+5 V dc $\pm$ .25 V
Pin 28 of U105 Socket	+12 V dc $\pm$ .6 V
Pin 1 of U51 Socket	+5 V dc $\pm$ .25 V
Pin 2 of U51 Socket	-12 V dc $\pm$ .6 V

\*All voltages referenced to ground.

- ( ) If any voltages are incorrect, locate and correct the cause before going on to Step 37.
- ( ) If voltages are correct, turn power supply off, disconnect power cable and go on to Step 37.
- ( ) Step 37. Install the following IC's in the indicated locations. Pay careful attention to the proper orientation.

NOTE

Dots on the assembly drawing and PC board indicate the location of pin 1 of each IC.

(Step 37 continued on Page III-28.)

# PROCESSOR TECHNOLOGY CORPORATION

Sol-PC SINGLE BOARD TERMINAL COMPUTER<sup>TM</sup>

SECTION III

<u>IC NO.</u>	<u>TYPE</u>
( ) U51*	TMS6011NC*
( ) U105* #	8080, 8080A or 9080A* #

\*MOS device. Refer to CAUTION on Page III-8.

#Note that pin 1 of this IC is in the upper left corner.

( ) Step 38. Perform Functional Test No. 1 of CPU circuits.

( ) Set S1 switches as follows:

No. 1 through 5: OFF

No. 6: ON

( ) Turn monitor on and apply power to Sol-PC.

( ) Momentarily ground pin 1 of U2. You should see a full display (64 characters x 16 lines) on the monitor.

( ) Momentarily ground pin 2 of U75. The display should blank while pin 2 of U75 is grounded. When you remove the ground, the display shown in Figure 3-9 on Page III-29 should appear.

## NOTE

The pattern shown in Figure 3-9 (delete characters) results from all bits of the DIO Bus being high. If you do not see the delete characters, one or more bits of the DIO bus are low. Consult the MCM6575 or MCM6574 pattern, as appropriate, in Section VIII of this manual to determine which bits are low.

( ) If the test fails, determine and correct the cause before proceeding with assembly.

( ) If the Sol-PC passes this test, turn monitor and power supply off, disconnect power cable and proceed to Step 39.

( ) Step 39. Install the following IC's in the indicated locations. Pay careful attention to the proper orientation.

(Step 39 continued on Page III-29.)



NOTE

Dots on the assembly drawing and PC board indicate the location of pin 1 of each IC.

<u>IC NO.</u>	<u>TYPE</u>
( ) U80	8T97#
( ) U81	8T97#
#DO NOT substitute.	

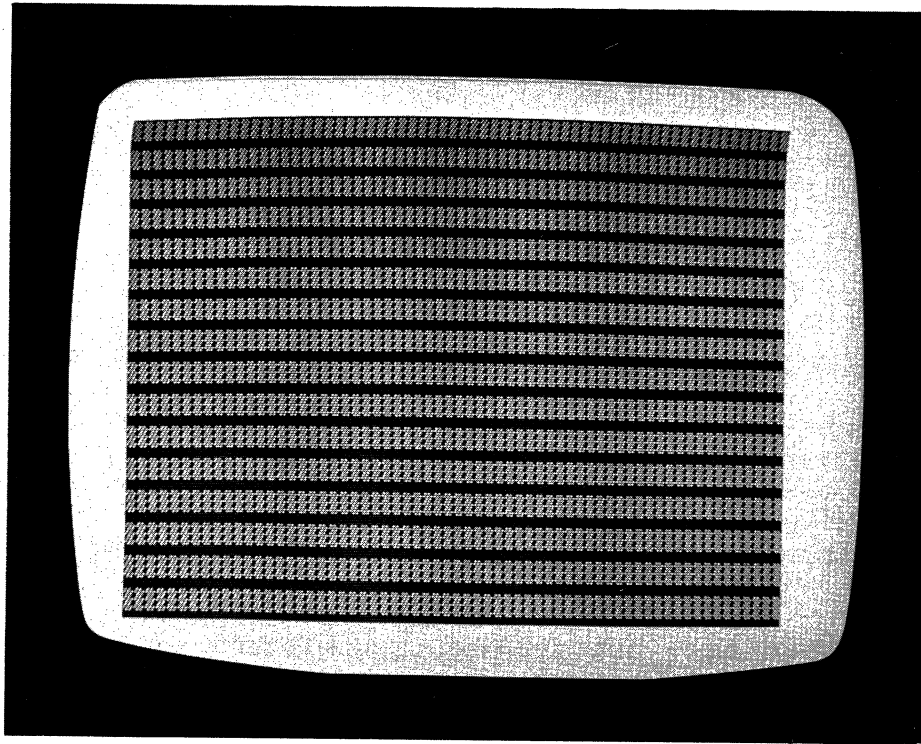


Figure 3-9. CPU Functional Test No. 1 display, 6574 or 6575 character generator (U25).

( ) Step 40. Perform Functional Test No. 2 of CPU circuits.

( ) Check that S1 switches are set as specified in Step 38.

( ) Turn monitor on and apply power to Sol-PC.

( ) Momentarily ground pin 1 of U2 and pin 2 of U75. The display shown in Figure 3-10 on Page III-31 should appear on the monitor.

( ) If the test fails, determine and correct the cause before proceeding with assembly.

( ) If the Sol-PC passes this test, turn monitor and power supply off, disconnect power cable and proceed to Step 41.

## PROCESSOR TECHNOLOGY CORPORATION

Sol-PC SINGLE BOARD TERMINAL COMPUTER<sup>TM</sup>

## SECTION III

- ( ) Step 41. Install the following IC's in the indicated locations. Pay careful attention to the proper orientation.

NOTE

Dots on the assembly drawing and PC board indicate the location of pin 1 of each IC.

<u>IC NO.</u>	<u>TYPE</u>
( ) U65	74LS253
( ) U66	74LS253
( ) U78	74LS253
( ) U79	74LS253
( ) U93	74LS175
( ) U106	74LS175
( ) U70	74LS109

- ( ) Step 42. Turn monitor on, apply power to Sol-PC and perform the test described in Step 40, except ground pin 5 of U75 instead of pin 2. You should get the same results.
- ( ) If the test fails, determine and correct the cause before proceeding with assembly.
- ( ) If the Sol-PC passes this test, turn monitor and power supply off, disconnect power cable and proceed to Step 43.
- ( ) Step 43. Install the following resistors in the indicated locations. Bend leads to fit distance between mounting holes, insert leads, pull down snug to board, solder and trim.

<u>LOCATION</u>	<u>VALUE (ohms)</u>	<u>COLOR CODE</u>
( ) R13	1.5K	brown-green-red
( ) R14	1.5K	" " "
( ) R15	1.5K	" " "
( ) R60	1.5K	" " "

- ( ) Step 44. Using two 4-40 x 5/8 binder head screws, two #4 insulating washers, two lockwashers and hex nuts, install 30-pin right angle edge connector in location J5. Insert screws from back (solder) side of board and place an insulating washer on each screw on front (component) side of board. Position connector with socket side facing right, place over screws and seat pins in mounting holes. Then place lockwasher on each screw, start nuts and tighten. Solder pins to board.

- ( ) Step 45. Using four 4-40 x  $\frac{1}{4}$  binder head screws, lockwashers and hex nuts, install two brackets (Sol-1040) for personality module in area to right of J5. Position brackets over the mounting holes as shown in Figure 3-11. Insert screws from front (component) side of board, place lockwasher on each screw on back (solder) side of board, start nuts and tighten.

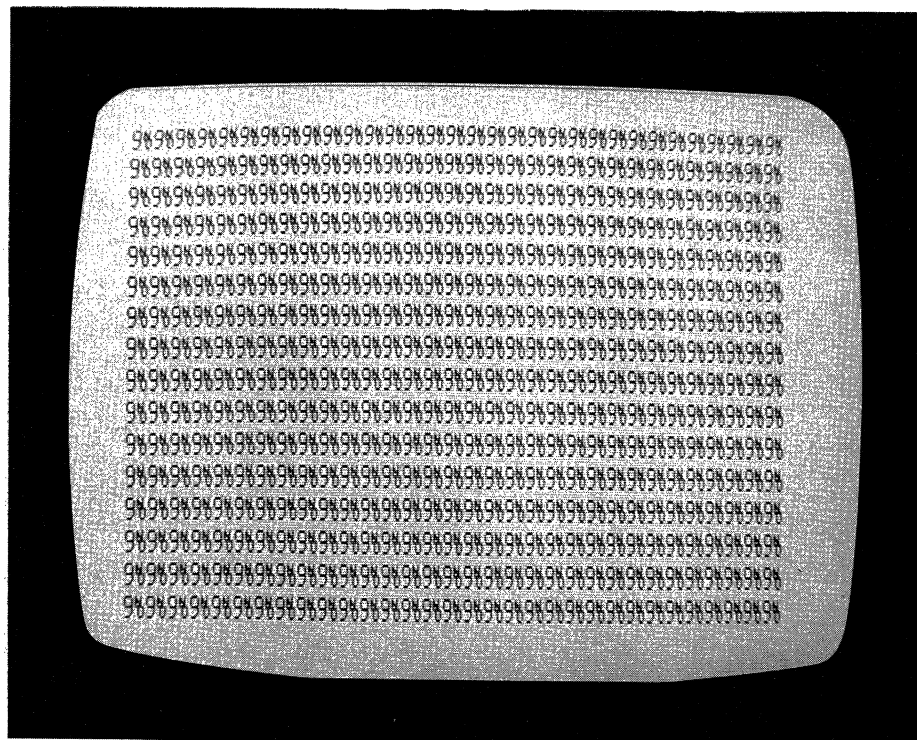


Figure 3-10. CPU Functional Test No. 2 display, 6575 character generator (U25).  
6574 displays: 9 □ 9 □ 9 □ etc.

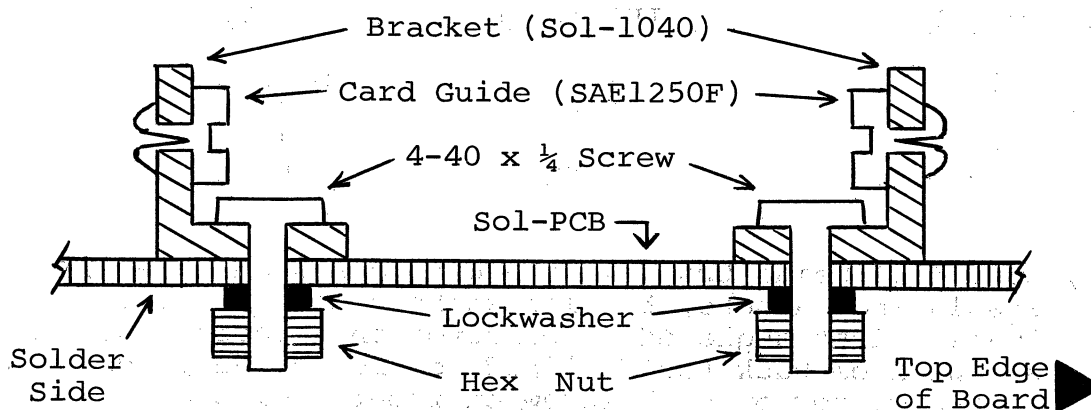


Figure 3-11. Personality module bracket/guide installation (Viewed from right end of Sol-PCB).

- ( ) Step 46. Attach plastic card guide (SAE1250F) to each of the brackets installed in Step 45. (See Figure 3-11.) Insert posts on guides into bracket holes and push in until they snap into place.
- ( ) Step 47. Install the following IC's in the indicated locations. Pay careful attention to the proper orientation.

NOTE

Dots on the assembly drawing and PC board indicate the location of pin 1 of each IC.

<u>IC NO.</u>	<u>TYPE</u>
( ) U3*	91L02APC or 2102L1PC*
( ) U4*	91L02APC or 2102L1PC*
( ) U5*	91L02APC or 2102L1PC*
( ) U6*	91L02APC or 2102L1PC*
( ) U7*	91L02APC or 2102L1PC*
( ) U8*	91L02APC or 2102L1PC*
( ) U9*	91L02APC or 2102L1PC*
( ) U10*	91L02APC or 2102L1PC*
( ) U22	74LS136
( ) U23	74LS20
( ) U24	74LS04
( ) U34	74LS138
( ) U35	74LS138
( ) U36	74LS138
( ) U53	74LS02 or 9LS02
( ) U71	74LS367
( ) U83	74LS20

\*MOS device. Refer to CAUTION on Page III-8.

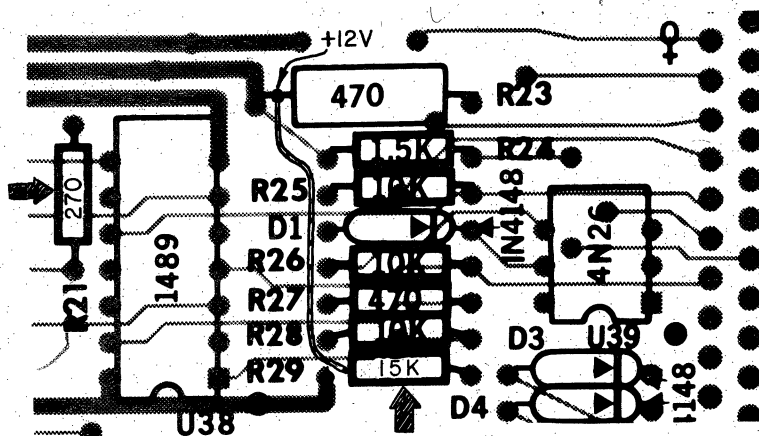
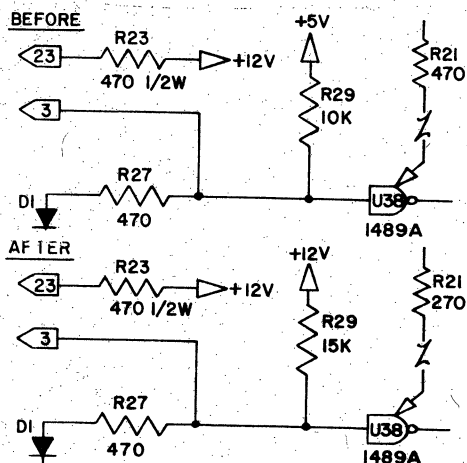
- ( ) Step 48. Test memory and decoder circuits.
    - ( ) Set S1 switches as specified in Step 38.
    - ( ) Turn monitor on and apply power to Sol-PC.
    - ( ) Ground pin 1 of U2. You should see the same display as shown in Figure 3-10 on Page III-31. In this case, however, there should be a vertical "flickering" movement with an apparent flicker rate of approximately three times per second.
    - ( ) Turn Switch No. 1 of S1 to ON. The flicker should stop.
- (Step 48 continued on Page III-33.)

## CHANGE NOTICE #10

Refer to Section X, Drawing X-17, Serial Data Interface/U.A.R.T. block. The section of U38 which has its input connected to pin 3 of J1 (Serial Loop Current Source) may not have enough drive at its input under worst case conditions, with the present values of R29, the input pull-up resistor, and R21, the resistor connected to pin 12 of U38. Substitute a 15K, 1/4 watt, 5% resistor for R29 (shown as 10K), and a 270 ohm, 1/4 watt, 5% resistor for R21 (shown as 470 ohm). These parts are included in the kit. Note in the schematics below that R29 is to be returned to +12 instead of +5. When R29 is installed, put the .9" length of tubing over the right hand lead. Clip this lead .1" longer than the tubing. With the legend on the P.C. board in normal reading position, hook this lead around the left-hand lead of R 24, a 1.5K resistor, and solder. Inspect the solder joint and lead dress for shorts. An assembly drawing detail of this modification is shown below.

Make the following changes in the manual before assembling Sol-PC.

Step	Page No.	Figure No., if any	Changes
1	X-17	Schematic, Input/Output	Change R21 value to 270 ohm
2	X-17	"	Change R29 value to 15K
3	X-17	"	Change R29 return to +12 V
4	X-3	Sol-PC Rev E Assembly	Change R21 value to 270 ohm
5	X-3	"	Change R29 value to 15K
6	X-3	"	Change R29 return to +12 V
7	III-3	Sol-PC Parts List	Add 1 270 ohm 1/4 watt, 5%
8	III-3	"	Change Qty. 470 ohm 1/4 watt to 2
9	III-3	"	Change Qty. 10K ohm to 31
10	III-3	"	Change Qty. 15K ohm to 2
11	III-33	Step 50	Change R21: 270 ohms, red-violet-brown
12	III-33	"	Change R29: 15K ohms, brown-green-orange
13	III-33	"	Under Step 50 instructions add: "See Change Notice #10"



Serial Comm. Port





## PROCESSOR TECHNOLOGY CORPORATION

Sol-PC SINGLE BOARD TERMINAL COMPUTER<sup>TM</sup>

## SECTION III

- ( ) If the test fails, determine and correct the cause before proceeding with assembly.
- ( ) If the Sol-PC passes this test, turn monitor and power supply off, disconnect power cable, set Switch No. 1 of S1 to OFF and go on to Step 49.
- ( ) Step 49. Assemble personality module if you have not yet done so. (See Section IV.) If you have, go to Step 9 in Section IV and complete the personality module assembly.
- ( ) Step 50. Install the following resistors in the indicated locations. Bend leads to fit distance between mounting holes, insert leads, pull down snug to board, solder and trim.

<u>LOCATION</u>	<u>VALUE (ohms)</u>	<u>COLOR CODE</u>
( ) R21	470 <del>270</del>	<sup>RED Violet Br</sup> yellow-violet-brown
( ) R22	470, $\frac{1}{2}$ watt	" " "
( ) R23	470, $\frac{1}{2}$ watt	" " "
( ) R24	1.5K	brown-green-red
( ) R25	10 K	brown-black-orange
( ) R26	10 K	" " "
( ) R27	470	yellow-violet-brown
( ) R28	10 K	brown-black-orange
( ) R29	10K	" <del>GREEN</del> "
( ) R37	1.5K	brown-green-red
( ) R38	1.5K	" " "
( ) R39	5.6K	green-blue-red
( ) R40	1.5K	brown-green-red
( ) R42	1.5K	" " "
( ) R43	1.5K	" " "
( ) R44	1.5K	" " "
( ) R45	330	orange-orange-brown
( ) R46	5.6K	green-blue-red
( ) R47	10 K	brown-black-orange
( ) R48	10 K	" " "
( ) R49	1.5K	brown-green-red
( ) R59	1.5K	" " "
( ) R61	1.5K	" " "
( ) R62	5.6K	green-blue-red
( ) R63	5.6K	" " "
( ) R64	330	orange-orange-brown
( ) R65	330	" " "
( ) R66	330	" " "
( ) R67	330	" " "
( ) R68	330	" " "
( ) R69	330	" " "
( ) R70	330	" " "
( ) R71	330	" " "

(Step 50 continued on Page III-34.)

<u>LOCATION</u>	<u>VALUE (ohms)</u>	<u>COLOR CODE</u>
( ) R72	680	blue-gray-brown
( ) R73	680	" " "
( ) R74	680	" " "
( ) R75	680	" " "
( ) R76	680	" " "
( ) R77	680	" " "
( ) R78	680	" " "
( ) R79	680	" " "
( ) R92	5.6K	green-blue-red
( ) R93	1.5K	brown-green-red
( ) R94	10 K	brown-black-orange
( ) R95	15 K	brown-green-orange
( ) R116	1.5K	brown-green-red

- ( ) Step 51. Install the following capacitors in the indicated locations. Take care to observe the proper value and type for each installation. Bend leads outward on solder (back) side of board, solder and trim. (Refer to NOTE in Step 2.)

<u>LOCATION</u>	<u>VALUE</u>	<u>TYPE</u>
( ) C29	.1 ufd	Disc
( ) C30	330 pfd	Disc

- ( ) Step 52. Install diodes D1 (1N4148 or 1N914), D2 (1N4001) and D3 through D6 (1N4148 or 1N914) in their locations in the area of U39. Position all diodes with their dark band (cathode) to the right.

- ( ) Step 53. Install the following DIP switches in the indicated locations. Take care to observe proper orientation.

<u>LOCATION</u>	<u>TYPE</u>	<u>ORIENTATION</u>
( ) S2	8-position	Switch No. 1 at top
( ) S3	8-position	Switch No. 1 at top
( ) S4	6-position	Switch No. 1 at top

- ( ) Step 54. Install Q1 (2N2907 or 2N3460) in its location between U55 and U56. The emitter lead (closest to tab on can) is oriented toward the bottom and the base lead toward the right. Push straight down on transistor until it is stopped by the leads. Solder and trim.

- ( ) Step 55. Using two 4-40 x 7/16 binder head screws, hex nuts and lockwashers, install 25-pin female connector in location J1 (serial I/O interface). Position connector with socket side facing right and insert pins into their holes in the circuit board. Insert screws from back (solder) side of board, place lockwasher on each screw, start nuts and tighten. Then solder connector pins to board.

- ( ) Step 56. Using two 4-40 x 7/16 binder head screws, hex nuts and lockwashers, install 25-pin male connector in location J2 (parallel I/O interface). Install J2 in the same manner as you did J1.
- ( ) Step 57. Install Augat pins in mounting holes K, L and M. (Refer to "Installing Augat Pins" in Appendix IV.) These holes are located between U85 and U86. No jumper will be installed.
- ( ) Step 58. Install the following IC's in the indicated locations. Pay careful attention to the proper orientation.

NOTE

Dots on the assembly drawing and PC board indicate the location of pin 1 of each IC.

<u>IC NO.</u>	<u>TYPE</u>
( ) U37	74LS367
( ) U38*	1489A*
( ) U39#	4N26#
( ) U52	74LS109
( ) U55	74LS00
( ) U56	1458CP or 1558CP
( ) U57	7406
( ) U58	8T94
( ) U72	74LS109
( ) U73	74LS109
( ) U84*	4029*
( ) U85*	4046*
( ) U86*	4024*
( ) U95	74173 3T10
( ) U96	74173
( ) U97	74175

\*MOS device. Refer to CAUTION on Page III-8.

#Solder this IC in its location. See "Loading DIP Devices" in Appendix IV.

- ( ) Step 59. Check input/output (I/O) circuits.

NOTE

The parallel I/O interface should be tested with the device you will be using. Refer to "I/O Interfacing" in Section VII.

(Step 59 continued on Page III-36.)

To check the serial I/O circuits, proceed as follows:

- ( ) Set S1 as in previous test,  
Set S2 switches all OFF,  
Set S3 switches all OFF, except S3-1 ON,  
Set S4 switches all OFF
- ( ) Set all S4 switches to OFF.
- ( ) Connect Sol-PC video output cable to monitor, turn monitor on and apply power to Sol-PC.
- ( ) Set Sol-PC to local by depressing LOCAL key on keyboard to turn keyboard indicator light on.
- ( ) Data entered from the keyboard should appear on the monitor.
- ( ) If the Sol-PC fails this test, locate and correct the cause before proceeding.
- ( ) If the Sol-PC passes this test, turn monitor and power supply off, disconnect power cable and video output cable and go on to Step 60.
- ( ) Step 60. Install the following resistors in the indicated locations. Bend leads to fit distance between mounting holes, insert leads, pull down snug to board, solder and trim.

<u>LOCATION</u>	<u>VALUE (ohms)</u>	<u>COLOR CODE</u>
( ) R117	10 K	brown-black-orange
( ) R118	10 K	" " "
( ) R119	10 K	" " "
( ) R139	1.0M	brown-black-green
( ) R140	10 K	brown-black-orange
( ) R141	150 K	brown-green-yellow
( ) R142	10 K	brown-black-orange
( ) R143	1 M	brown-black-green
( ) R144	47 K	yellow-violet-orange
( ) R145	10 K	brown-black-orange
( ) R146	10 K	" " "
( ) R147	2.2M	red-red-green
( ) R148	100 K	brown-black-yellow
( ) R149	100	brown-black-brown
( ) R150	470	yellow-violet-brown
( ) R151	5.6K	green-blue-red
( ) R152	150 K	brown-green-yellow
( ) R153	100 K	brown-black-yellow
( ) R154	100 K	yellow " " "OK"
( ) R155	6.8, ½ watt	blue-grey-gold
( ) R156	6.8, ½ watt	blue-grey-gold
( ) VR3	50 K	Potentiometer



- ( ) Step 61. Install the following capacitors in the indicated locations. Take care to observe the proper value and type for each installation. Bend leads outward on solder (back) side of board, solder and trim. (Refer to NOTE in Step 2.)

CAUTION

REFER TO FOOTNOTE AT END OF THIS STEP  
BEFORE INSTALLING C67.

<u>LOCATION</u>	<u>VALUE (ufd)</u>	<u>TYPE</u>
<del>C47</del>	.001	Disc
<del>C48</del>	.047	"
<del>C49</del>	.001	"
<del>C50</del>	.01	Mylar Tubular
<del>C51</del>	.1	Disc
<del>C66</del>	.1	"
<del>C67*</del>	1	Tantalum
<del>C68</del>	.1	Disc
<del>C69</del>	.1	"
<del>C70</del>	.1	"
<del>C71</del>	.001	"
<del>C72</del>	.001	Mylar Tubular
<del>C73</del>	.047	Disc
( ) C74	470 pfd	"

\*Install C67 with "+" lead at top right.

- ( ) Step 62. Install miniature phone jacks in locations J6 and J7 located to the right of U101. Position J6 and J7 with jack facing right, insert pins in mounting holes and solder.
- ( ) Step 63. Install subminiature phone jacks in locations J8 and J9 in lower right corner of board. Install J8 and J9 as you did J6 and J7.
- ( ) Step 64. Install Q3 (2N4360) in its location to the left of C67. Install Q3 with its flat "side" at the bottom. Push straight down on transistor until it is stopped by the leads, solder and trim.

CAUTION

THE 2N4360 IS STATIC SENSITIVE. REFER TO  
CAUTION ON PAGE III-8.

- ( ) Step 65. Install Q4 and Q5 (2N2222) in their locations above and to the left of U108. For both transistors, the emitter lead (closest to tab on can) is oriented toward the left and the base lead toward the right. Push straight down on transistor until it is stopped by the leads, solder and trim.

- ( ) Step 66. Install diodes D13 and D14 (1N4001) in their locations in the lower right corner of the board. Position both diodes with their dark band (cathode) at the bottom.
- ( ) Step 67. Install DIP reed relays in locations K1 and K2 to the right of U113. Be sure to install K1 and K2 with their end notch at the bottom (pin 1 in lower right corner). These relays are soldered to the board. (Refer to "Loading DIP Devices" in Appendix IV.)
- ( ) Step 68. Install the following IC's in the indicated locations. Pay careful attention to the proper orientation.

NOTE

Dots on the assembly drawing and PC board indicate the location of pin 1 of each IC.

<u>IC NO.</u>	<u>TYPE</u>
( ) U69*	TMS6011NC*
( ) U98*	4023*
( ) U99*	4030*
( ) U100*	4013*
( ) U101*	4027*
( ) U108	1458CP or 1558CP
( ) U109*	4049*
( ) U110*	4046*
( ) U111*	4019*
( ) U112*	4520*
( ) U113*	4013*

\*MOS device. Refer to CAUTION on Page III-8.

- ( ) Step 69. Install Augat pins in mounting holes H, I and J (located to left of C70). (Refer to "Installing Augat Pins" in Appendix IV.) Using #24 bare wire, install a jumper between pins I and J.
  - ( ) Step 70. Adjust VR3.
    - ( ) Using a cable with a male phono jack on both ends, connect ACI audio output (J6) to ACI audio input (J7).
    - ( ) Apply power to Sol-PC.
    - ( ) Set VR3 fully clockwise (CW).
    - ( ) Measure the DC voltage at pin 13 of U110 and write the measured voltage down. (Call this Voltage A.)
    - ( ) Set VR3 fully counterclockwise (CCW).
- (Step 70 continued on Page III-39.)

## PROCESSOR TECHNOLOGY CORPORATION

Sol-PC SINGLE BOARD TERMINAL COMPUTER<sup>TM</sup>

## SECTION III

- ( ) Measure the DC voltage at pin 13 of U110 and write the measured voltage down. (Call this Voltage B.)
- ( ) Add Voltages A and B and divide the sum by 2. (Call the result Voltage C.) An example follows:

Voltage A (VR3 full CW): 3.45 V dc  
 Voltage B (VR3 full CCW): 1.80 V dc

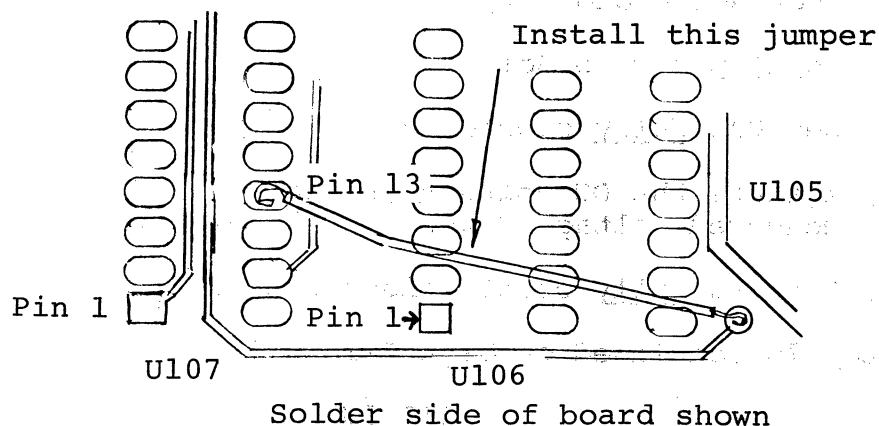
$$A + B = 5.25 \text{ V dc}$$

$$\text{Voltage C} = 5.25 \text{ V dc} / 2 = 2.63 \text{ V dc}$$

- ( ) Adjust VR3 so that the voltage at pin 13 of U110 equals Voltage C. (In the preceding example this would be 2.63 V dc.)
- ( ) Step 71. If your recorder has only a microphone input, remove the I-to-J jumper you installed in Step 69 and install a jumper (#24 bare wire is recommended) between the I and H pins.

Otherwise, leave the I-to-J jumper in and go on to Step 72.

- ( ) Step 72. Install 100-pin edge connector, J11. Using two 4-40 x 7/16 binder head screws, install 100-pin edge connector in location J11 (center of PC board). Seat the pins in the mounting holes. Then thread screws from front (component) side of board into the threaded inserts that are pre-installed in the J11 mounting holes. Tighten screws and solder pins to board.
- ( ) Step 73. Look on the rear of the board, on the component side, where the Personality Module plugs in, for a mark "Rev E". If your board is marked this way, complete this step, otherwise ignore this step. Connect a jumper of #24 a.w.g. insulated wire between pin 13 of U107 and the feed-through pad adjacent to pin 21 of U105. Solder, check for solder bridges, and trim excess wire strands if needed. The installed jumper is shown below.



## 3.6.4 Modification for 625 Line Video

The European televisions standard defines a raster of 625 lines at a field rate of 50 Hz. The horizontal rate of the U.S. standard, 15,750 Hz., is maintained. Only the number of scan lines on the screen is increased.

The Video Display Generator section may be modified for the 50 Hz. standard by following the additional steps below. The effect of the modification is to increase the modulus of the counter U62 to eight during VDISP. This results in four extra character lines (52 scan lines) between the bottom and top of the display area, for a total of 312 scan lines per field and 624 scan lines per frame.

The field rate should be close enough to 50 Hz. to reduce any swim effects to less than 0.1 Hz. Some difficulty may be encountered in obtaining centering of the display within the frame. This is because the stand-off time to VSYNC from the bottom of the display is unchanged from the 60 Hz. standard. If objectionable, increase the value of resistor R100 which is in series with the VPOS control.

To convert for 50 Hz., perform these additional steps:

- ( ) Locate U62 on the component side legend. Find pin 5 of this IC on the component (front) side of the board. Cut the "V"-shaped trace connecting pin 5 to the nearby pad designated "AF", using a sharp exacto blade or scribe, so that there is no continuity between these pads.
- ( ) Bend a small piece of bare wire, such as a resistor clipping, into a loop to form a jumper between pad "AF", and the adjacent pad "AG". Insert the jumper, pull close to the board, solder, and trim the leads.

If this modification is made, change the schematic, X-18, to show that pin 5 of U62 now connects to pin 4 (ground), instead of pin 6 as shown.



#### IV PERSONALITY MODULE ASSEMBLY

4.1	Parts and Components . . . . .	IV-1
4.2	Assembly Tips . . . . .	IV-1
4.3	Assembly Precautions . . . . .	IV-1
4.4	Required Tools, Equipment and Materials . .	IV-1
4.5	Orientation . . . . .	IV-1
4.6	Assembly-Test . . . . .	IV-2
4.6.1	Circuit Board Check . . . . .	IV-2
4.6.2	Assembly-Test Procedure . . . . .	IV-2







## Sol PERSONALITY MODULE

## SECTION IV

## 4.1 PARTS AND COMPONENTS

When ordering your Sol, you selected one of two types of Personality Modules: CONSOL Or SOLOS. The outer carton of your kit is stamped with the Personality Module type. Both use the same PC board marked 2708, assembly #107000, and differ only in the type of ROM's and their programming. (An alternative PC board marked 5204 and designed for type 5204 EPROM's is also available but not supplied with this kit. Schematic diagram X-4 and assembly drawing X20 refer to this alternative board.) Check all parts against Table 4-1 below. If you have difficulty identifying any parts, refer to Figure 3-1 on page III-5. One of two kits, using the same PC board: 2708-0 or 2708-1 may be supplied. The 2708-0 version uses one 9216 masked ROM which has no window on top of the IC package. The 2708-1 version uses two 2708 EPROM's which have windows.

Table 4-1. PM2708 Personality Module Parts List.

1	PM2708 PC Board	1 or 4*	1-ufd Capacitor,
1 or 2*	9216 ROM or 2708 EPROM's		Tantalum Dipped
	with Personality program	1 or 2*	24-pin DIP Socket
1	74LS08	1	14-pin DIP Socket
0 or 2*	1N5231B Zener Diode	1	Handle Bracket (Sol-1045)
3 or 4*	10K ohm, $\frac{1}{4}$ watt, 5% Res.	2	2-56X1/8 Binder Head
0 or 2*	100 ohm, $\frac{1}{2}$ watt, 5% Res.		Screw
1	.047-ufd Disc Ceramic		

\* These are the quantities of parts used in the 2708-1 version.

## 4.2 ASSEMBLY TIPS

For the most part the assembly tips given in Paragraph 3.2 of Section III (Page III-1) apply to assembling the personality module.

## 4.3 ASSEMBLY PRECAUTIONS

For the most part the assembly precautions given in Paragraph 3.3 in Section III (Page III-6) apply.

## 4.4 REQUIRED TOOLS, EQUIPMENT AND MATERIALS

The following tools, equipment and materials are recommended for assembling the personality module.

- |                            |                            |
|----------------------------|----------------------------|
| 1. Needle nose pliers      | 5. 60-40 rosin-core solder |
| 2. Diagonal cutters        | (supplied)                 |
| 3. Screwdriver             | 6. Small amount of #24     |
| 4. Soldering iron, 25 watt | solid wire                 |

## 4.5 ORIENTATION

Capacitor location C2 will be located in the upper left hand corner of the board when the edge connector is positioned at the

left end of the board. In this position the component (front) side of the board is facing up. Subsequent position references related to the personality module circuit board assume this orientation.

#### 4.6 ASSEMBLY-TEST

##### 4.6.1 Circuit Board Check

- ( ) Visually check circuit board for broken traces, shorts (solder bridges) between traces and similar defects.
- ( ) Check circuit board to insure that the +5-volt bus, +12 volt bus and -12-volt bus are not shorted to each other or to ground. Using an ohmmeter, make the following measurements (refer to personality module assembly drawing in Section X):
  - ( ) +5 volt Bus Test. On U1, measure between pin 12, (ground) and pin 24 (+5 volts). There should be no continuity.
  - ( ) -5 volt Bus Test. On U1 and U2, measure between pin 12 (ground) and pin 21 (-5 volts). There should be no continuity.
  - ( ) +12 volt Bus Test. Also on U1, measure between pin 12 (ground) and the bottom edge connector pin on the component side of the board marked A1.
  - ( ) Inter-bus Test. On U1, measure between pins 12 and 21, then between edge connector pin A1 and pins 21, then 12. There should be no continuity in any of these measurements.

If visual inspection reveals any defect, or you measure continuity in any of the preceding tests, return the board to Processor Technology for replacement. If the board is not defective, proceed to next paragraph.

##### 4.6.2 Assembly-Test Procedure

Refer to personality module assembly drawing X-6.

#### CAUTION

THE MEMORY IC'S USED ON THE PERSONALITY MODULE ARE MOS DEVICES. THEY CAN BE  
(CAUTION continued on Page IV-3)

DAMAGED BY STATIC ELECTRICITY DISCHARGE. HANDLE THESE IC's SO THAT NO DISCHARGE FLOWS THROUGH THE IC. AVOID UNNECESSARY HANDLING AND WEAR COTTON, RATHER THAN SYNTHETIC, CLOTHING WHEN HANDLING MOS IC's. (STATIC DISCHARGE PROBLEMS ARE MUCH WORSE IN LOW HUMIDITY CONDITIONS.)

- ( ) Step 1. Install DIP sockets. Install each socket in the indicated location with its end notch oriented as shown on the circuit board and assembly drawing. Take care not to create solder bridges between the pins and/or traces.

INSTALLATION TIP

Insert socket pins into mounting pads of appropriate location. On back (solder) side of board, bend pins at opposite corners of socket (e.g. pins 1 and 9 on a 16-pin socket) outward until they are at a 45° angle to the board surface. This secures the socket until it is soldered. Repeat this procedure with each socket until all are secured to the board. Then solder the pins on all sockets.

<u>LOCATION</u>	<u>TYPE SOCKET</u>
( ) U1	24 pin
( ) U2*	24 pin*
( ) U3	14 pin

\*Used on 2708-1 version only.

- ( ) Step 2. Install the following resistors in the indicated locations. Install these resistors parallel with the board. Bend leads by using needle nose pliers to grip the resistor lead right next to the resistor body, and bend the portion of the lead on the other side of the pliers with your finger. The bend must be the right distance from the resistor body for the resistor to fit easily into its two holes. Insert the leads into the two holes, and from the opposite side of the board pull the leads to bring the resistor body down to touch the board. Bend the leads outward on the solder (back) side of the board so the resistors do not slip out of position.

<u>LOCATION</u>	<u>VALUE</u>	<u>COLOR CODE</u>
( ) R1*	100 ohms	brown-black-brown
( ) R2*	100 ohms	brown-black-brown
( ) R3	10K	brown-black-orange
( ) R4*	10K	brown-black-orange
( ) R5	10K	brown-black-orange
( ) R6	10K	brown-black-orange

\*not used on 2708-0 version

( ) Step 3. Install 1N5231B Zener Diodes in locations Z1, and Z2 if you have the 2708-1 version. Form the leads as in Step 2. Insert the diodes so that the white band on the diode is in the position indicated by the legend. Bend the leads outward to retain the diodes, then solder and trim the leads.

( ) Step 4. Install the following capacitors in the indicated locations. Take care to observe the proper value, type and orientation for each installation. On the dipped tantalum capacitors, the "+" lead is the one which is closest to the "+" marking on the body of the capacitor. Insert this lead in the hole marked "+" on the PC board legend. After inserting C5, remove it from the board before soldering to clear wax from the leads and holes. After inserting all capacitors, pull them close to the board and bend the leads outward to secure them. Solder and trim all leads.

<u>LOCATION</u>	<u>VALUE (ufd)</u>	<u>TYPE</u>
( ) C1*	1	Dipped Tantalum
( ) C2	1	Dipped Tantalum
( ) C3*	1	Dipped Tantalum
( ) C4*	1	Dipped Tantalum
( ) C5	.047	Disc Ceramic

\*not used on 2708-0 version

( ) Step 5. Check for +5, +12, and -12 volt bus-to-ground shorts. Using an ohmmeter on OHMS times 1K or OHMS times 10K scale, make the following measurements. A typical reading is 1 Megohm. A reading less than 10K indicates a short.

- ( ) Measure between edge connector pins A2 and A15.
- ( ) Measure between edge connector pins A14 and A15.
- ( ) Measure between edge connector pins A1 and A15.
- ( ) If any measurement indicates a short, find and correct the problem before proceeding.

Rev C ( ) Step 6. Using two 2-56 x 1/8" binder head screws, install

handle bracket (Sol-1045). Position bracket on front (component) side of board at the right end as shown in Figure 4-2. Align bracket holes with mounting holes in board, insert screws from back (solder) side of board and drive into bracket. No nuts are needed since the bracket holes are tapped.

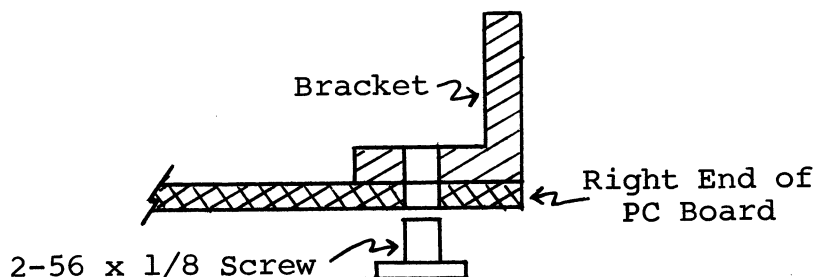


Figure 4-2. Handle bracket (Sol-1045) installation.

- ( ) Step 7. If you have a 2708-0 version with the 9216 ROM (windowless), omit this step. If you have the 2708-1 version, find the area above the U1 socket where the legend reads "-5V 21 CO 19 +12V." This legend designates five PC pads in a row directly underneath. On the back (solder) side of the board, there is a small trace which connects the "CO" and "21" pad. Cut this trace with a sharp knife or scribe point so there is no longer continuity between these pads. Form the clipping from a resistor lead, or other small bare wire into a loop and insert this jumper between the "-5V" pad and the "21" pad. Solder and trim the leads. Next find the two pads between C2 and R6, with legend "-16" under the right pad of the pair. On the back (solder) side of the board, cut the trace which connects these pads.
- ( ) Step 8. Stop assembly at this point and proceed with Sol-PC assembly and test up through Step 48. (See Section III.) Then go on to Step 9 of this procedure.
- ( ) Step 9. Plug personality module into J5 on Sol-PC, apply power to Sol-PC and make the following voltage measurements on the personality module, with respect to chassis ground:

<u>MEASUREMENT POINT</u>	<u>VOLTAGE</u>
Pin 24 of U1, U2	+5 V dc $\pm$ 5%
Pin 14 of U3	+5 V dc $\pm$ 5%
Pin 21*of U1, U2	-5 V dc $\pm$ 5%
Pin 12 of U1, U2	Ground
Pin 7 of U3	Ground

\*For 2708-1 version only

- ( ) Measure between edge connector pin B14 and pin B15. You should measure more than 1M ohms. A reading less than 10K ohms indicates a short.

## PROCESSOR TECHNOLOGY CORPORATION

## Sol PERSONALITY MODULE

## SECTION IV

- ( ) If any voltages are incorrect, locate and correct the cause before proceeding to Step 10.
- ( ) If the voltages are correct, turn power off, disconnect power cable, unplug personality module and go on to Step 10.
- ( ) Step 10. Install IC's in the sockets numbered U1 through U3. Make sure the dot or notch indicating pin 1 on the IC package is in the correct position as indicated on the PC board component legend and the assembly drawing X-6. Socket U2 is left empty on 2708-0 versions (9216 ROM with no window). As shown in the table, the 2708 EPROM's have paper labels with the designation shown, while 9216 ROM's have the designation printed on the IC package itself.

	IC NO.	TYPE	IC LABEL	
			CONSOL	SOLOS
2708-0 version	( ) U1*	2708	C	S4
	( ) U2*	2708	Empty	S5
	( ) U3	74LS08	--	--
2708-1 version	( ) U1*	9216	--	SOLOS
	( ) U2	Empty	--	--
	( ) U3	74LS08	--	--

\*MOS devices. See CAUTION on pages IV-2, 3.

- ( ) Step 11. Plug personality module into J5 on Sol-PC and connect Sol-PC video output cable to video monitor. (Refer to CAUTION on Page III-22 in Section III.)
- ( ) Set S1 switches as follows:
  - No. 1 through 4: OFF
  - No. 5: ON
  - No. 6: OFF
- ( ) Turn monitor on and apply power to Sol-PC
- ( ) With both the CONSOL and SOLOS modules, you should see the cursor, preceded by a prompt character, like this:
 

>■
- ( ) If you do not see a cursor, locate and correct the problem before proceeding.



- ( ) If a blinking cursor is present, the ENter and DUmP commands should operate as described in Section IX of this manual.
- ( ) If the ENter and DUmP commands do not operate correctly, locate and correct the problem before proceeding.
- ( ) If the personality module is operating correctly, turn monitor and power off, disconnect power cable and video output cable and go on to Step 50 in Section III. (The personality module can be left plugged in.)



## V KEYBOARD ASSEMBLY and TEST

5.1	Parts and Components . . . . .	V-1
5.2	Assembly Tips . . . . .	V-1
5.3	Assembly Precautions . . . . .	V-1
5.4	Required Tools, Equipment and Materials . .	V-1
5.5	Orientation . . . . .	V-1
5.6	Assembly-Test . . . . .	V-3
5.6.1	Circuit Board Check . . . . .	V-3
5.6.2	Assembly-Test Procedure . . . . .	V-3





### 5.1 PARTS AND COMPONENTS

Check all parts and components against the "Parts List", Table 5-1. If you have difficulty in identifying any parts by sight, refer to Figure 3-1 on Page III-5 in Section III of this manual.

### 5.2 ASSEMBLY TIPS

For the most part the assembly tips given in Paragraph 3.2 of Section III (Page III-1) apply to assembling the Sol keyboard.

In addition, be sure your hands are clean before handling the circuit board, especially the area containing the keyboard switch pads.

### 5.3 ASSEMBLY PRECAUTIONS

For the most part the assembly precautions given in Paragraph 3.3 in Section III (Page III-6) apply to assembling the Sol keyboard.

### 5.4 REQUIRED TOOLS, EQUIPMENT AND MATERIALS

The following tools, equipment and materials are recommended for assembling the personality module:

1. Needle nose pliers
2. Diagonal cutters
3. Screwdriver (thin blade)
4. Controlled heat soldering iron, 25 watt
5. 60-40 rosin-core solder (supplied)

### 5.5 ORIENTATION

Light emitting diode location LED3 will be located in the lower left-hand corner of the board when locations J1 and U4 through U16 are at the top of the board. In this position the component (front) side of the board is facing up and all horizontal reading legends will read from left to right. Subsequent position references related to the keyboard circuit board assume this orientation.

Table 5-1. Sol Keyboard Parts List.

<u>INTEGRATED CIRCUITS</u>		
1 555 (U3)	1 74LS30 (U25)	
1 2101 or 9101 (U20)	2 7442 (U17 & 21)	
2 4051A (U19 & 22)	5 74LS74 (U8,9,11,15,26)	
4 74LS00 (U4,10,14,16)	2 7493 (U6,U5)	
1 74LS04 (U23)	1 74LS132 (U7)	
1 7406 (U24)	2 74LS175 (U1,U2)	
2 74LS10 (U13 & 27)	1 8334, 9334 or 83L34 (U12)	
	1 8574, 74S287, or 82S129 (U18)	
<u>TRANSISTORS</u>	<u>DIODES (ZENER)</u>	<u>DIODES (LIGHT EMITTING)</u>
6 2N3640	1 1N5221B (D1)	3 MV5752 (LED1,2,3)
3 2N4274		
<u>RESISTORS</u>	<u>CAPACITORS</u>	
1 10 ohm, $\frac{1}{4}$ watt, 5%	2 220 pfd, disc	
3 150 ohm, $\frac{1}{4}$ watt, 5%	1 470 pfd, disc	
1 390 ohm, $\frac{1}{4}$ watt, 5%	1 .0022 ufd, disc	
1 680 ohm, $\frac{1}{4}$ watt, 5%	2 .01 ufd, disc	
7 1 K ohm, $\frac{1}{4}$ watt, 5%	5 .047 ufd, disc	
10 1.5K ohm, $\frac{1}{4}$ watt, 5%	1 .1 ufd, Mylar tubular	
1 2.2K ohm, $\frac{1}{4}$ watt, 5%	2 15 ufd, tantalum dipped	
5 3 K ohm, $\frac{1}{4}$ watt, 5%		
2 33 K ohm, $\frac{1}{4}$ watt, 5%		
2 68 K ohm, $\frac{1}{4}$ watt, 5%		
2 2.2K ohm resistor network		
2 33 K ohm resistor network		



Table 5-1. Sol Keyboard Parts List (Continued).

MISCELLANEOUS

- 1 Sol-KBD Printed Circuit Board
- 1 8-pin DIP Socket
- 17 14-pin DIP Socket
- 8 16-pin DIP Socket
- 1 22-pin DIP Socket
- 1 20-pin Header, 3M3492-2002
- 1 9-3/4" 20-conductor Rainbow Cable Assembly
- 1 70-key (Sol-10) or 85-key (Sol-20) Keyboard Assembly
- 1 Plastic Insert (Sol-10) for Key Pad
- 18 Torx Screw (Similar to #4 by 3/8" sheet metal screws.)
- 3 Fiber Spacer
- 1 Length Solder

## 5.6 ASSEMBLY-TEST

## 5.6.1 Circuit Board Check

- ( ) Visually inspect circuit board for obvious flaws. (The design of the board includes numerous unconnected traces and traces that are shorted to each other.)
- ( ) Check circuit board to insure that the +5-volt bus is not shorted to ground. Using an ohmmeter, measure between the GND and +5V pads located in the upper left corner of the board. There should be no continuity.

If no visual inspection reveals any defect, or you measure continuity between the GND and +5V pads, return the board to Processor Technology for replacement. If the board is not defective, proceed to next paragraph.

## 5.6.2 Assembly-Test Procedure

Refer to keyboard assembly drawing X-7.

CAUTION

SOME MOS INTEGRATED CIRCUITS ARE USED ON THE Sol KEYBOARD. THEY CAN BE DAMAGED BY STATIC ELECTRICITY DISCHARGE. HANDLE MOS IC's SO THAT NO DISCHARGE FLOWS THROUGH THE IC. AVOID UNNECESSARY HANDLING AND WEAR COTTON, RATHER THAN SYNTHETIC, CLOTHING WHEN YOU DO HANDLE MOS IC's. (STATIC CHARGE PROBLEMS ARE MUCH WORSE IN LOW HUMIDITY CONDITIONS.)

- ( ) Step 1. Install DIP sockets. Install each socket in the indicated location with its end notch oriented as shown on the circuit board and assembly drawing. Take care not to create solder bridges between the pins and/or traces. (Refer to "Installation Tip" on Page III-9 in Section III.)

<u>LOCATION</u>	<u>TYPE SOCKET</u>
( ) U1 and 2	16 pin
( ) U3	8 pin
( ) U4 through U11	14 pin
( ) U12	16 pin
( ) U13 through U16	14 pin
( ) U17 through U19	16 pin
( ) U20	22 pin
( ) U21 and 22	16 pin
( ) U23 through U27	14 pin

- ( ) Step 2. Install the following capacitors in the indicated locations. Take care to observe the proper value, type and orientation (if applicable) for each installation. Insert leads, pull down snug to board, bend leads outward on solder (back) side of board, solder and trim.

NOTE

Disc capacitor leads are usually coated with wax during the manufacturing process. After inserting leads through mounting holes, remove capacitor and clear the holes of any wax. Reinsert and install.

<u>LOCATION</u>	<u>VALUE</u>	<u>TYPE</u>	<u>ORIENTATION</u>
( ) C1	15 ufd	Tantalum	"+" lead top
( ) C2	.047 ufd	Disc	None
( ) C3	.1 ufd	Mylar	"

(Continued on Page V-5.)

## PROCESSOR TECHNOLOGY CORPORATION

## Sol1 KEYBOARD

## SECTION V

<u>LOCATION</u>	<u>VALUE</u>	<u>TYPE</u>	<u>ORIENTATION</u>
( ) C4	.01 ufd	Disc	None
( ) C5	.047 ufd	"	"
( ) C6	.047 ufd	"	"
( ) C7	.0022 ufd	"	"
( ) C8	470 pfd	"	"
( ) C9	220 pfd	"	"
( ) C10	220 pfd	"	"
( ) C11	.01 ufd	"	"
( ) C12	.047 ufd	"	"
( ) C13	.047 ufd	"	"
( ) C14	15 ufd	Tantalum	"+" lead top

- ( ) Step 3. Install the following resistors in the indicated locations. Bend leads to fit distance between mounting holes, insert leads, pull down snug to board, solder and trim.

<u>LOCATION</u>	<u>VALUE (ohms)</u>	<u>COLOR CODE</u>
( ) R1	150	brown-green-brown
( ) R2	150	" " "
( ) R3	150	" " "
( ) R4	68 K	blue-gray-orange
( ) R5	560 K	green-blue-yellow
( ) R6	33 K	orange-orange-orange
( ) R7	1 K	brown-black-red
( ) R8	1.5K	brown-green-red
( ) R9	3 K	orange-black-red
( ) R10	3 K	" " "
( ) R11	3 K	" " "
( ) R12	3 K	" " "
( ) R13	1.5K	brown-green-red
( ) R14	1.5K	" " "
( ) R15	1.5K	" " "
( ) R16	1 K	brown-black-red
( ) R17	390	orange-white-brown
( ) R18	1 K	brown-black-red
( ) R19	10	brown-black-black
( ) R20	1 K	brown-black-red
( ) R21	1 K	" " "
( ) R22	3 K	orange-black-red
( ) R23	1 K	brown-black-red
( ) R24	1 K	" " "
( ) R25	1.5K	brown-green-red
( ) R26	680	blue-gray-brown
( ) R27	33 K	orange-orange-orange
( ) R28	1.5K	brown-green-red
( ) R29	1.5K	" " "

(Continued on Page V-6.)

- | <u>LOCATION</u> | <u>VALUE (ohms)</u> | <u>COLOR CODE</u> |
|-----------------|---------------------|-------------------|
| ( ) R30         | 1.5K                | brown-green-red   |
| ( ) R31         | 1.5K                | " " "             |
| ( ) R32         | 68 K                | blue-gray-orange  |
| ( ) R33         | 1.5K                | brown-green-red   |
| ( ) R34         | 2.2K                | red-red-red       |
- ( ) Step 4. Install Zener diode D1 (1N5221B) in its location to the left of R17. Position D1 with its dark band (cathode) at the bottom.
- ( ) Step 5. Install Q1, Q2 and Q9 (2N4274) and Q3 through Q8 (2N3640) in their respective locations at the top center of the board. The emitter lead (closest to flat side of case) is oriented toward the right of the board and the base is oriented toward the top. Insert leads until transistor is approximately 3/16" above surface of circuit board, solder and trim.
- ( ) Step 6. Install resistor networks RX1 and RX3 (2.2K ohms) and RX2 and RX4 (33K ohms) in their respective locations just above the keyboard pads. Install each network so that the dot on its package is positioned next to the foil square on the circuit board. Recheck values before soldering.

CAUTION

THESE RESISTOR NETWORKS ARE DELICATE.  
HANDLE WITH CARE.

- ( ) Step 7. Install light emitting diodes LED1, 2 and 3 (MV5752) in their respective locations in the lower left corner of the circuit board. Insert leads through fiber spacer, position each diode with its cathode lead (longer lead and/or the lead next to flat edge of LED package) at the bottom, insert leads into mounting holes in circuit board, pull down so that spacer and LED are snug to board, solder and trim. (If fiber spacers are not supplied with your kit, install LED's so they are approximately 3/16" above surface of circuit board.)
- ( ) Step 8. Install 20-pin header in location J1 (upper left corner of board). Position header so pin 1 is in the lower left corner. (An arrow on the header points to pin 1.) Solder.
- ( ) Step 9. Using an ohmmeter, measure between GND and +5V pads in upper left corner of the board. You should measure some resistance. Zero resistance indicates a short. If required, find and correct the problem before proceeding to Step 10.

## PROCESSOR TECHNOLOGY CORPORATION

Sol KEYBOARD

SECTION V

- ( ) Step 10. Install the following IC's in the indicated locations. Pay careful attention to the proper orientation.

NOTE

Dots on the assembly drawing and PC board indicate the location of pin 1 of each IC.

<u>IC NO.</u>	<u>TYPE</u>
( ) U1	<sup>25</sup> 74LS175
( ) U2	74LS175
( ) U3	555
( ) U4	74LS00
( ) U5	7493
( ) U6	7493
( ) U7	74LS132
( ) U8	74LS74
( ) U9	74LS74
( ) U10	74LS00
( ) U11	74LS74
( ) U12	8334, 9334 or 83L34
( ) U13	74LS10
( ) U14	74LS00
( ) U15	74LS74
( ) U16	74LS00
( ) U17	7442
( ) U18	8574, 74S287, or 82S129
( ) U19*	4051A*
( ) U20*	2101 or 9101*
( ) U21	7442
( ) U22*	4051A*
( ) U23	74LS04
( ) U24	7406
( ) U25	74LS30
( ) U26	74LS74
( ) U27	74LS10

\*MOS device. Refer to CAUTION on Page V-4.

- ( ) Step 11. Connect 20-conductor ribbon cable between J1 on keyboard to J3 on Sol-PC so that cable goes left from J3.

- ( ) Step 12. Check keyboard operation.

( ) Set S1 switches on Sol-PC as follows:

No. 1 through 4: OFF

No. 5: ON

No. 6: OFF

- ( ) Connect TV monitor to Sol-PC.
- ( ) With personality module installed, apply power to Sol-PC.
- ( ) Using a CLEAN finger, touch key pad #62 (MODE SELECT).
- ( ) You should get a carriage return and line feed and see a "greater than" sign ( > ) on the screen above the cursor.

NOTE

You may have to touch pad #62 several times to obtain the specified display.

- ( ) If you are unable to obtain the specified display, locate and correct the problem before proceeding.
- ( ) If the keyboard is operating correctly, turn monitor and Sol-PC power off, disconnect 20-conductor ribbon cable at J1 on the keyboard and go on to Step 13.
- ( ) Step 13. Place keyboard assembly carefully over key pads on PC board. Be sure the three LED's fit in the holes in the sheet metal. Carefully align holes in PC board, 18 in all, with threaded mounting holes on bottom of keyboard assembly. Insert Torx screws from solder (back) side of board and, using a thin-blade screwdriver, drive into keyboard assembly mounting holes. Drive screws evenly and tighten just enough to hold keyboard assembly in place.

CAUTION: DO NOT OVERTIGHTEN THESE SCREWS.

- ( ) Step 14. Reconnect 20-conductor ribbon cable to J1 on keyboard.
- ( ) Step 15. Test keyboard for proper operation.
  - ( ) Apply power to monitor and Sol-PC.
  - ( ) Strike MODE SELECT key.
  - ( ) Strike UPPER CASE key. Indicator light should come on.
  - ( ) Strike UPPER CASE key again. Indicator light should go off.
  - ( ) Strike LOCAL key. Indicator light should come on.
  - ( ) Strike LOCAL key again. Indicator light should go off.



(Step 15 continued.)

- ( ) Strike SHIFT LOCK key. Indicator light should come on.
- ( ) Strike either SHIFT key. Indicator light should go off.
- ( ) Verify operation of all alphanumeric keys. (As you strike each key you should observe the corresponding character on the monitor.)
- ( ) Should the keyboard fail any of the preceding checks, locate and correct the problem before proceeding.
- ( ) If the keyboard passes all of the preceding tests, congratulations on a job well done.

At this point you have successfully assembled the Sol keyboard and tested it for proper operation. It is now ready for use with the Sol-PC Single Board Terminal Computer<sup>TM</sup>.

Having completed the Sol keyboard, power supply, Sol-PC and personality module, you are now ready to assemble the Sol cabinet-chassis. Cabinet-chassis assembly instructions are provided in Section VI.



VI Sol CABINET-CHASSIS ASSEMBLY

6.1	Introduction . . . . .	VI-1
6.2	Parts and Components . . . . .	VI-1
6.3	Assembly Tips . . . . .	VI-1
6.3.1	General . . . . .	VI-1
6.3.2	Electrical . . . . .	VI-1
6.3.3	Mechanical . . . . .	VI-5
6.4	Required Tools, Equipment and Materials . .	VI-6
6.5	Orientation . . . . .	VI-6
6.5.1	Sol Backplane Board, Sol-BPB . . . .	VI-6
6.5.2	Sol Cabinet-Chassis . . . . .	VI-6
6.6	Assembly-Test . . . . .	VI-7
6.6.1	Backplane Board (Sol-BPB) Assembly .	VI-7
6.6.2	Wooden-Masonite Parts . . . . .	VI-8
6.6.3	Sol Assembly . . . . .	VI-10





## 6.1 INTRODUCTION

This section covers assembly of the Sol-10 and Sol-20 chassis and cabinet. The instructions contained herein assume that you have already assembled the power supply and Sol-PC Single Board terminal Computer<sup>TM</sup>...including the personality module and the Sol keyboard (Sol-KBD).

## 6.2 PARTS AND COMPONENTS

Check all parts and components against the appropriate "Parts List(s)", Table 6-1 and 6-2. If you have any difficulty in identifying any parts by sight, refer to Figures 6-1 and 6-2 on Pages VI-4 and VI-4.

## 6.3 ASSEMBLY TIPS

### 6.3.1 General

1. Scan Section VI in its entirety before you start to assemble your Sol cabinet-chassis.

2. IT IS IMPORTANT that you follow the step-by-step instructions in the order given when assembling the Sol cabinet-chassis if your assembly is to be done correctly and with minimum effort.

3. Assembly steps and component installations are preceded by a set of parentheses. Check off each installation and step as you complete them. This will minimize the chances of omitting a step or component.

4. Should you encounter any problem during assembly, call on us for help if needed.

### 6.3.2 Electrical

1. Use a low-wattage soldering iron, 25 watts maximum, for all soldering.

2. Solder neatly and as quickly as possible.

3. Use only 60-40 rosin-core solder. NEVER use acid-core solder or externally applied fluxes.

4. DO NOT press the tip of the soldering iron on pads or traces when installing components and/or attaching leads to a PC board. To do so can cause the pad or trace to "lift" off the board and permanently damage it.

## PROCESSOR TECHNOLOGY CORPORATION

## Sol CABINET-CHASSIS

## SECTION VI

Table 6-1. Sol-10 Cabinet-Chassis Parts List.

<u>CHASSIS and SUBCHASSIS</u>	<u>HARDWARE</u>
1 Main Chassis	4 4-40 x 3/16 Screw, Machine
1 Expansion Chassis	16 4-40 x 5/16 Screw, Machine
1 Power Supply Subchassis	8 4-40 Hex Nut
1 Fan Closure Plate	22 #4 Lockwasher, Internal Tooth
<u>BRACKETS</u>	16 6-32 x 1/2 Screw, Machine
	8 6-32 Hex Nut
	16 #6 Lockwasher, Internal Tooth
1 Power Supply Subchassis Bracket	11 8-32 x 1/2 Screw, Machine
2 Keyboard Bracket	2 8-32 x 1 Screw, Machine
<u>CABINET</u>	3 8-32 Hex Nut
	3 #8 Lockwasher, Internal Tooth
	2 10-24 Thumb Screw
	21 #6 x 1/4 Screw, Sheet Metal
	4 #6 x 5/16 Screw, Sheet Metal
	12 5/8 Screw, Wood
	2 #4 Solder Lug
<u>MISCELLANEOUS</u>	10 Tinnerman Plastic Inserts, Tapped
	2 1/4" Spacer, 4-40 Tapped
	4 Self-stick Protective Pads
2 Finger Well Label, Black *	
1 Printed Trim Plate, Paper	
1 Plexiglass Strip	
1 Serial Number Label	
1 Connector Identification Label	

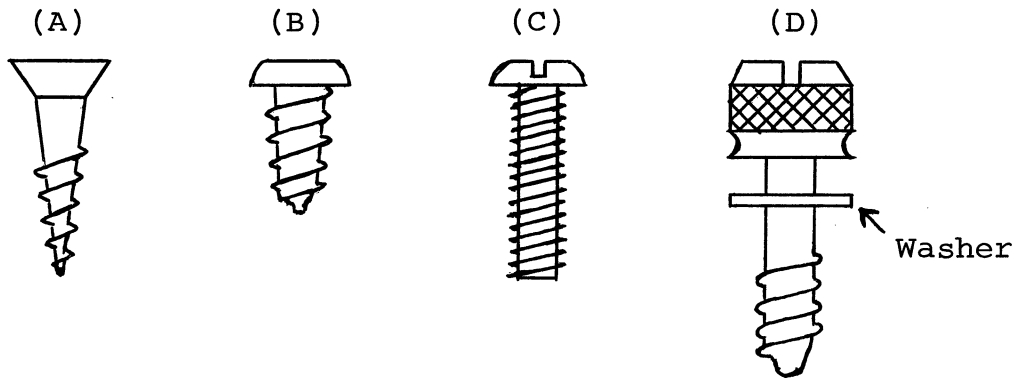
\* May be packaged under the plexiglass strip.



Table 6-2. Sol-20 Cabinet-Chassis Parts List.

The Sol-20 Cabinet-Chassis Kit includes all Sol-10 parts listed in Table 6-1 plus the following items:

- 1 Sol-BPB Circuit Board (Backplane Board)
- 1 3" 5-wire Cable with Molex Connector
- 1 100-pin Edge Connector, VIKING 3KH50/90V5
- 5 100-pin Edge Connector, other brand
- 2 Right Angle Backplane Bracket
- 1 Gusset Bracket, Left
- 1 Gusset Bracket, Right
- 10 Plastic Card Guide
- 6 4-40 x 5/16 Screw, Machine
- 6 4-40 x 5/8 Screw, Machine
- 12 4-40 Hex Nut
- 12 #4 Lockwasher, Internal Tooth
- 8 6-32 x  $\frac{1}{2}$  Screw, Machine
- 8 6-32 Hex Nut
- 9 #6 Lockwasher, Internal Tooth
- 12 #6 x  $\frac{1}{4}$  Screw, Sheet Metal



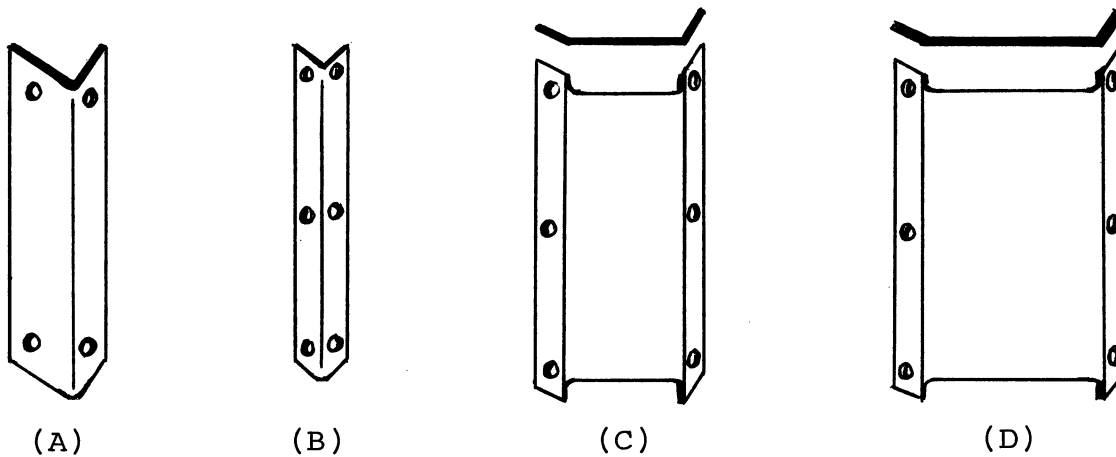
(A) Flat Head Wood Screw

(C) Binder or Pan Head Screw

(B) Sheet Metal Screw

(D) Thumb Screw

Figure 6-1. Types of screws used in Sol cabinet-chassis assembly.



(A) Keyboard Bracket

(B) Backplane Bracket, Right Angle

(C) Gusset Bracket, Left

(D) Gusset Bracket, Right

(E) Power Supply Subchassis Bracket

(E)

Figure 6-2. Brackets used in Sol cabinet-chassis assembly.

## 6.3.2 Electrical (continued)

5. (Sol-20 only.) The Backplane PC board (Sol-BPB) has plated-through holes. Solder flow through to the component side of the board can produce solder bridges (shorts). Check for such bridges after you install each component or wire.

6. (Sol-20 only.) The Backplane PC board (Sol-BPB) has an integral solder mask (a lacquer coating) that shields selected areas on the board. This mask minimizes the chances of creating solder bridges during assembly.

## 6.3.3 Mechanical

1. If you do not have the proper screwdrivers (see Paragraph 6.5), we recommend that you buy them rather than using a knife point, a blade screwdriver on a Phillips screw, and other makeshift means. Proper screwdrivers minimize the chances of stripping threads, disfiguring screw heads and marring decorative surfaces.

2. To assure a correct fit and tight assembly, be sure you use the screws specified in the instructions.

3. Lockwashers are widely used in the Sol cabinet-chassis assembly so that screws will not loosen when subjected to stress or vibration. When a lockwasher is specified, do not omit it and make sure you install it correctly.

4. Some instructions call for prethreading holes. This is done to make assembly easier by giving you maximum working space for installing relatively hard-to-drive sheet metal screws. If you bypass prethreading instructions you will only make your cabinet-chassis assembly more difficult.

To prethread a hole, insert specified screw in the hole and position it as straight as possible. While holding the screw in this position, drive it into the metal with the proper screwdriver. If started straight the screw will continue to go straight into the metal so that the head and sheet metal surfaces are in full contact.

5. The diameter of the shank (threaded portion) of a screw increases in relation to its number. For example, a 6-32 screw is larger in diameter than a 4-40 screw. Also, a #8 lockwasher is larger than a #4 lockwasher.

## 6.4 REQUIRED TOOLS, EQUIPMENT AND MATERIALS

The following tools, equipment and materials are recommended for assembling the Sol cabinet-chassis. (Unless indicated otherwise, none of the following items are supplied with your kit.)

1. Needle nose pliers
2. Diagonal cutters
3. Screwdriver, thin  $\frac{1}{4}$ " blade
4. Screwdriver, #2 Phillips
5. Controlled heat soldering iron, 25 watt
6. 60-40 rosin-core solder (supplied)
7. Silicon grit abrasive paper, 220 & 400 grit
8. Boiled linseed oil
9. Turpentine or mineral spirits
10. Masking tape
11. Transparent tape
12. Rubber mallet or small hammer

## 6.5 ORIENTATION

## 6.5.1 Sol Backplane Board, Sol-BPB (Sol-20 Only)

The PC board identification ( Sol-BPB) and revision level will be located in the upper left-hand corner of the board when the edge connector (gold contacts) is positioned at the bottom of the board. In this position, the component (front) side of the board is facing up. Subsequent position references related to the Sol-BPB assume this orientation.

## 6.5.2 Sol Cabinet-chassis

Unless specified otherwise, all position references (e.g., left, right, front, back, bottom and top) in the cabinet-chassis assembly instructions assume the Sol cabinet is viewed from the front (keyboard) when it is sitting in its normal position (key-board up).

## 6.6 ASSEMBLY-TEST

NOTE

Instructions that apply only to the Sol-20 are preceded by an asterisk. Skip these instructions if you are assembling a Sol-10.

## \*6.6.1 Backplane Board (Sol-BPB) Assembly

Refer to assembly drawing, page X-11.

- \*( ) Step 1. Visually inspect Sol-BPB PC board for obvious flaws such as solder bridges (shorts) between traces, broken traces and similar defects.

If visual inspection reveals any defects, return the board to Processor Technology for replacement. If the board passes inspection, go on to Step 2.

- \*( ) Step 2. Install VIKING 3KH50/9VC5 100-pin edge connector on top edge of PC board. (This edge has silver (not gold) contacts.)

NOTE

This connector is supplied as a trouble-shooting aid. It is not critical to normal operation of the Sol-20.

Position connector on PC board so that its #1 trace is aligned with the #1 trace on the board, and push connector fully onto board. Bend the connector pins slightly so that both rows of pins are in light contact with the traces on the board. DO NOT CLOSE CONNECTOR PINS SO MUCH THAT YOU WILL DAMAGE THE TRACES WHEN PLACING THE CONNECTOR OVER THE EDGE OF THE BOARD. While holding the connector and board together, place board solder side down on a book, or other flat surface that is higher than your work surface, so the connector extends fully over the edge. That is, the connector should not rest on the book. Reposition connector if needed to align the pins and traces. On the component (front) side of board, solder a pair of traces. On the component (front) side of board, solder a pair of pins at each end of the connector to their respective traces on the

(Step 2 continued on Page VI-8.)

board. Then solder the remaining 46 pins on the component side to traces.

The connector must be perpendicular to the edge of the board. If it is not, bend the pins you just soldered to obtain the required alignment. Then solder the other 50 pins to the traces.

\* ( ) Step 3. Install the other five 100-pin edge connectors. Position connector on front side of board and insert pins. On solder (back) side of board, solder pins at opposite corners of the connector to hold it in place. Then solder remaining 98 pins. (Refer to Paragraph 6.6.1 on Page VI-6 for definition of front side of board.)

\* ( ) Step 4. Connect 3" 5-wire cable to circuit board to upper-most pads in top right corner: Insert wires from solder (back) side of board and solder on component (front) side of board. If a wire is too large for the mounting hole, snip off as many individual strands as needed to obtain a fit. Connect cable leads as follows:

CABLE LEAD

PAD

White	Ground (fifth hole from right)
White	Ground (fourth hole from right)
Blue	+8 V dc (third hole from right)
Red-White	+16 V dc (second hole from right)
Yellow-White	-16 V dc (first hole from right)

NOTE

Pad orientations given above are as viewed from component (front) side of circuit board.

\* ( ) Step 5. Fill all exposed (not covered with lacquer) feed-through holes on right-hand side of board with solder.

The backplane board is now assembled. Set it to one side for later installation in the cabinet-chassis.

#### 6.6.2 Wooden-masonite Parts

Refer to assembly drawings, pages X8 and X9.

( ) Step 6. Finish walnut side panels.

The sides of the Sol cabinet are solid black walnut which have been sanded to a smooth surface. If there should be any blemishes, remove them with 220 grit abrasive paper. SAND WITH THE GRAIN...NEVER ACROSS THE GRAIN.



We recommend an oil finish be applied to the walnut since such a finish lies "in" the wood, not on "top" of it. Also, no wax is necessary with an oil finish.

You may obtain a good finish by using a half-and-half mixture of boiled linseed oil and turpentine. Apply mixture with rag, soaking all surfaces. (End grain may require more oil than face grain.) Let stand for one-half hour, recoating any dry spots, and wipe dry with a clean cloth. Repeat as often as needed to obtain a lustrous finish. (It may take several days.)

You may also use a commercially available penetrating oil such as Watco Danish Oil or Tung Oil. Follow directions on the container if you use such an oil. For a more durable finish when using a penetrating oil:

1. Sand between applications with 220 grit silicon carbide abrasive paper. (Wipe clean after 15 minutes to avoid build-up.)
  2. Repeat the following day using 400 grit paper between applications.
  3. Repeat as often as desired, using a still finer grit paper between applications. DO NOT sand after final application, but wipe the surfaces clean and let dry for one day. Then coat with previously mentioned linseed oil-turpentine mixture and wipe dry.
- ( ) Step 7. Using a black broad tipped felt pen, darken all edges of the two masonite parts.
- ( ) Step 8. Mate the left walnut and masonite side pieces. (Refer to assembly drawing on page X-8.)

#### NOTE

When the walnut and masonite side pieces are correctly mated, the countersink side of the six countersunk (funnel-shaped enlargement) holes in the masonite will be next to the main chassis.

Insert five Tinnerman plastic inserts in the holes indicated on Drawing X-8. Insert these from the side that mates with the walnut. These inserts may be seated by gently tapping them with a hammer until fully seated.

- ( ) Step 9. Insert remaining five Tinnerman inserts in right masonite side piece as you did the left side piece. (Refer to Drawing on page X-9.)

- ( ) Step 10. Attach left masonite side piece to left walnut side piece with six 5/8" flat head wood screws. Drive these screws through the countersunk holes in the masonite into the walnut. (Refer to Drawing No. X-8.)

NOTE

Lead holes have been predrilled in the walnut to make driving these screws easier.

- ( ) Step 11. Attach right masonite side piece to right walnut side piece as you did the left side pieces. (Refer to Drawing No X-9.)
- ( ) Step 12. Set both side piece assemblies to one side.

6.6.3 Sol Assembly

Refer to Drawing No. X-10 in Section X. Figure 6-3 and 6-4 show complete Sol assemblies without covers.

- ( ) Step 13. Mount keyboard support bracket (heavy gauge right angle brackets) to each side of the main chassis as shown in Drawing No. X-10. These are mounted with the narrower side of the bracket at the top.

Attach each bracket to main chassis with two 6-32 x 1/2 binder or pan head screws and #6 lockwashers. Place lockwasher on screw, insert screw from outer surface of main chassis side wall and drive into the threaded bracket mounting holes.

- ( ) Step 14. Attach power supply subchassis bracket (short leg "T" shaped bracket) to top front of power supply subchassis as shown in Drawing No. X-10. (Note that leg of "T" is closer to side wall of subchassis. This leg is for mounting a "power on" indicator light--not supplied.) Insert #6 x 1/4 sheet metal screw from right side of side wall and drive into bracket.
- ( ) Step 15. To gain access to the rear area of the power supply subchassis side wall, remove the #6 x 1/4 sheet metal screw that attaches the fan closure plate to the subchassis. You should not have to disconnect the transformer (black wires) or AC receptacle ground (green wire) leads since they have sufficient slack to permit moving the closure plate out of the way. (Set screw to one side for use in re-installing the fan closure plate.)

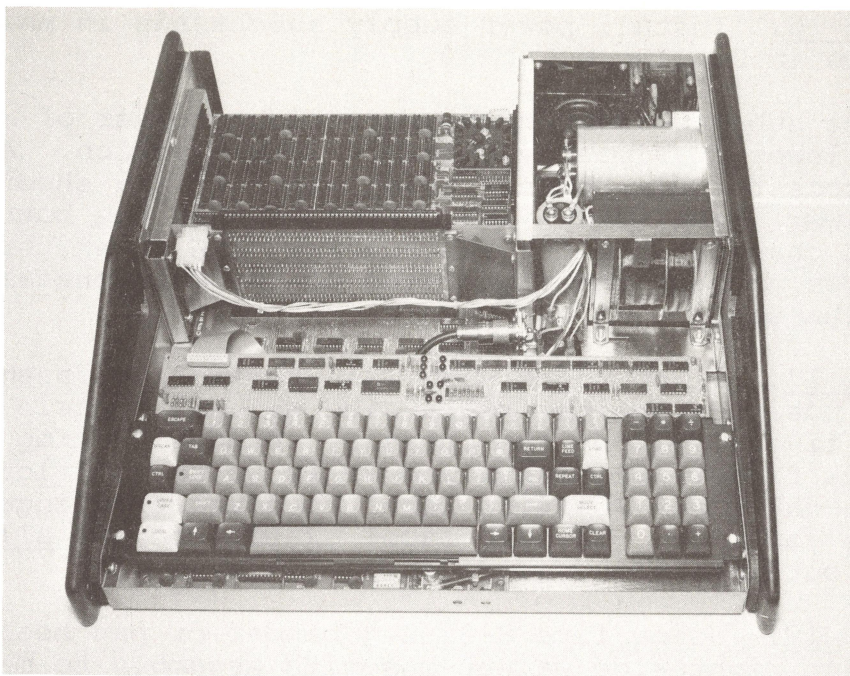


Figure 6-3. Sol-20 with covers removed. Front (or keyboard) is in foreground, power supply is in right rear corner, expansion chassis (with 8KRA Memory installed) is to left of power supply. The vertical board just behind white connector on left is the backplane board.

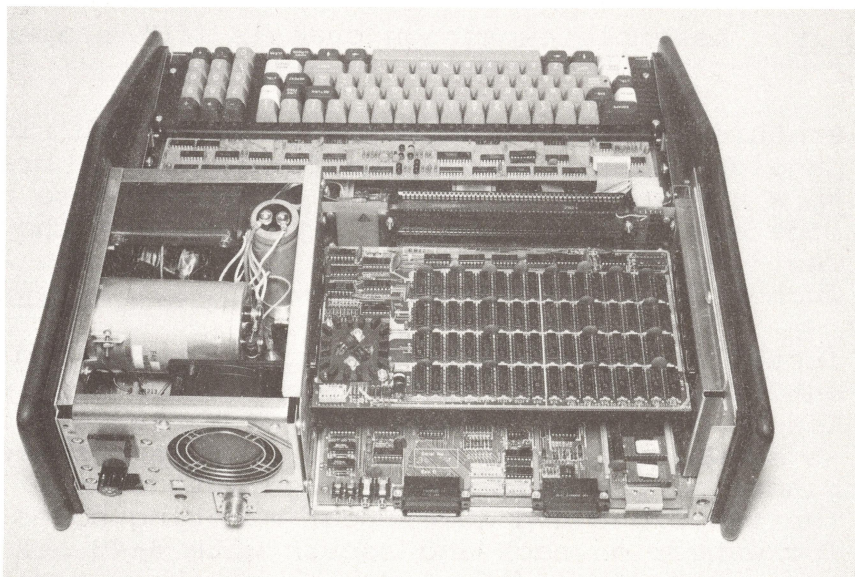


Figure 6-4. Sol-20 with covers removed. Rear side of assembly is in foreground and Sol-PC is just visible at lower right rear of assembly. 8KRA Memory is installed in expansion chassis above Sol-PC.

- ( / ) Step 16. Install power supply subchassis in main chassis as shown in Drawing No. X-10.

Place subchassis over the rear right corner of main chassis and lower it almost vertically into position. Attach subchassis to main chassis using eight #6 x  $\frac{1}{4}$  sheet metal screws. Five screws are driven through the bottom of the main chassis into the subchassis. The other three are driven through the right side of the main chassis into the subchassis.

- ( ) Step 17. Place right side walnut-masonite assembly in proper position against right side of main chassis and outline the finger well on the chassis. Remove backing from one black finger well label and affix it to the right side of main chassis. Position label to cover the finger well outline you made. Be sure label extends beyond all edges of the outline.

- ( ) Step 18. Using five 8-32 x  $\frac{1}{2}$  binder or pan head screws, attach right side walnut-masonite assembly to main chassis and power supply subchassis as shown in Drawing No. X-10. Insert screws from inside surface of chassis and drive into the plastic inserts you installed in Step 9. Note that the two front screws are driven through the main chassis, the two lower rear screws are driven through both the power supply subchassis, and the upper rear screw is driven through the power supply subchassis.

- ( ) Step 19. Assemble expansion chassis ("U" shaped chassis).

\* ( / ) Prethread 12 mounting holes (six on each side) on expansion chassis side walls for backplane brackets with #6 x  $\frac{1}{4}$  sheet metal screws. Three of these holes on each side are located near the front edge of the main chassis. The remaining three holes on each side are about  $1\frac{1}{2}$  to 2 inches behind the front three. Leave screws installed.

( / ) Install female coaxial connector on the tab that extends out from the lower right front of the expansion chassis. Insert connector through tab so threaded end faces left as shown in Drawing No. X-10. Insert three 4-40 x  $\frac{5}{16}$  binder or pan head screws from left side of tab through the two front and lower rear mounting holes. Place #4 lockwasher on each and secure with 4-40 hex nuts. Insert another 4-40 x  $\frac{5}{16}$  binder or pan head screw through upper rear mounting hole and install 4-40 hex nut. (Leave this nut loose.)

\* ( ) Install 10 plastic card guides (five on each side) on inside surface of both side walls of the expansion chassis.

These are installed over the ventillation grilles with the gripper fingers pointing towards the backplane board. To install, simply insert posts on guide into appropriate mounting holes and push in until they snap into place.

- ( ) Step 20. Install expansion chassis on main chassis as shown in Drawing No. X-10.

Position expansion chassis with coaxial connector at the front (near FWB3 on power supply subchassis) over left rear area of main chassis and lower into place. Attach expansion chassis to main chassis using eight #6 x  $\frac{1}{4}$  sheet metal screws. Four screws are driven through the bottom of the main chassis into the expansion chassis, three are driven through the left side of the main chassis into the expansion chassis, and one is driven through the lower left corner of the back side of the main chassis into the expansion chassis.

- ( ) Step 21. Attach left end of power supply subchassis bracket to expansion chassis as shown in Drawing No. X-10. Drive one 6 x  $\frac{1}{4}$  sheet metal screw through expansion chassis into bracket.

- ( ) Step 22. Route coaxial cable from connector on fan closure plate along left side of power supply subchassis to connector on expansion chassis.

- ( ) Step 23. Using the #6 x  $\frac{1}{4}$  sheet metal screw you removed in Step 15, re-attach fan closure plate to power supply subchassis. (Make sure side lip on plate is on right side of expansion chassis side wall.

- ( ) Step 24. Attach fan closure plate to expansion chassis with two #6 x  $\frac{1}{4}$  sheet metal screws. Drive screws through expansion chassis into fan closure plate.

#### NOTE

If lip on fan closure plate and expansion chassis are not in contact, insert one or two  $\frac{1}{2}$ " flat washers as needed between the two surfaces. Place washers so screws pass through them.

- ( ) Step 25. Connect free end of coaxial cable from connector on fan closure plate to connector on expansion chassis. Solder inner conductor to pin of connector. Remove hex nut on upper rear connector mounting screw, place lockwasher lug (coaxial shield) on screw and secure with nut.

- ( ) Step 26. Install male coaxial connector on free end of coaxial cable that is connected to Sol-PC (the composite video output cable). Install connector as follows (refer to Figure 6-5):

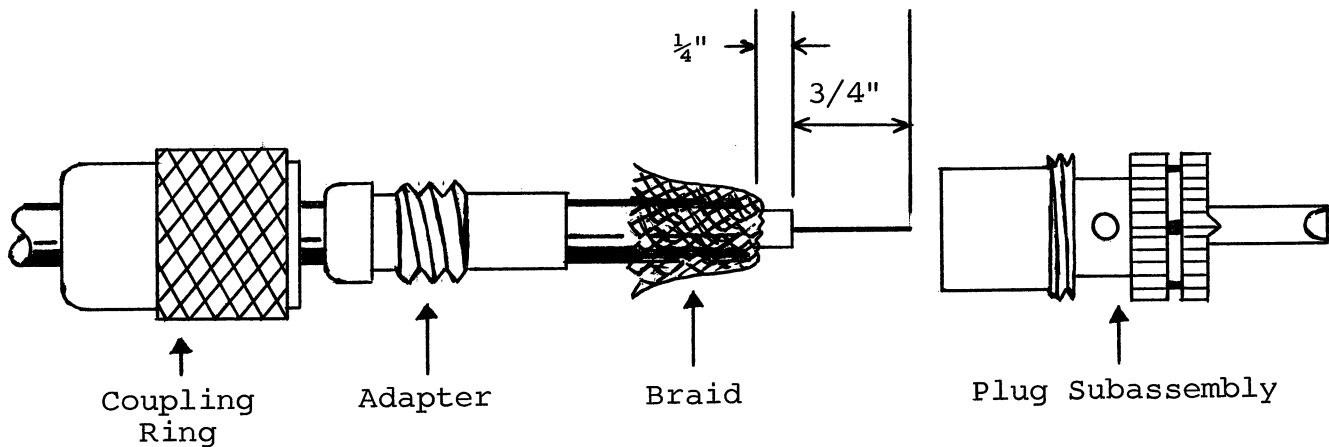


Figure 6-5. Sol-PC coaxial cable connector assembly.

- ( ) Slide coupling ring and adapter on cable in that order and cut end of cable even.
- ( ) Remove one inch of outer insulation.
- ( ) Fan braid slightly and fold back over outer insulation as shown.
- ( ) Slide adapter fully up under braid and press braid down over adapter body.
- ( ) Trim braid so that it does not interfere with adapter threads.
- ( ) Remove 3/4" of inner conductor insulation and tin exposed conductor.
- ( ) Slide cable fully into plug subassembly and screw subassembly on adapter.
- ( ) Solder braid to plug subassembly shell through solder holes. (Use enough heat to create a good bond between braid and shell.)
- ( ) Solder center conductor to plug contact by filling contact with solder. Cut off excess conductor.
- ( ) Slide coupling ring over plug subassembly and screw it onto plug.



- ( ) Step 27. Install Sol-PC in expansion chassis.

Position Sol-PC on bottom of expansion chassis with J1, J2 and J6 through J9 at the rear. Align threaded standoffs on bottom of Sol-PC with the oblong holes in the bottom of the main chassis.

Attach Sol-PC to chassis with eight 4-40 x 5/16 binder or pan head screws and #4 lockwashers. Place washer on screw and drive screw loosely into standoff from bottom of main chassis. Leave all eight screws loose.

- ( ) Step 28. Connect Sol-PC composite video output cable to expansion chassis coaxial connector.

- ( ) Step 29. Affix black finger well label to left side of main chassis in same manner as you did the right side. (See Step 17.) MAKE SURE LABEL DOES NOT OBSTRUCT ANY OF THE COOLING VENTS.

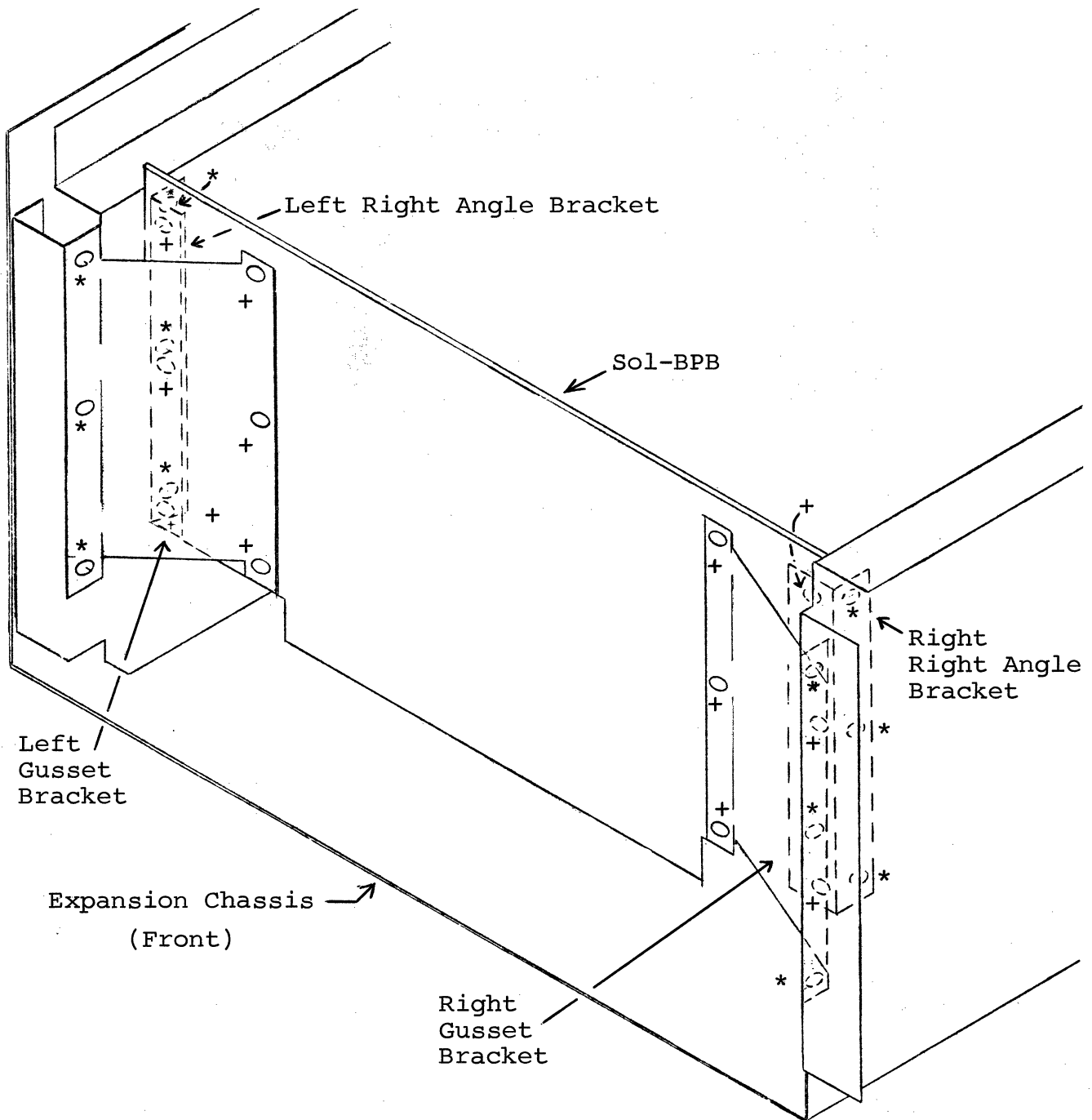
- ( ) Step 30. Using three 8-32 x  $\frac{1}{2}$  and two 8-32 x 1 binder or pan head screws, attach left side walnut-masonite assembly to main chassis as shown in Drawing No. 101000. Insert screws from inside surface of chassis and drive into the plastic inserts you installed in Step 8. Note that the two front screws (8-32 x  $\frac{1}{2}$ ) are driven through the main chassis, the uppermost screw (8-32 x  $\frac{1}{2}$ ) is driven through the expansion chassis, and the two lower rear screws (8-32 x 1) are driven through both the expansion chassis and main chassis.

- \* ( ) Step 31. Install left and right backplane right angle brackets (light gauge brackets) on expansion chassis side walls. Refer to Figure 6-6 on Page VI-16.) These two brackets are installed just to the front of the card guides and should be positioned as shown in Figure 6-6. Attach each bracket to the chassis with three #6 x  $\frac{1}{4}$  sheet metal screws. USE THE SCREWS YOU USED IN STEP 19 TO PRETHREAD THE HOLES.

- \* ( ) Step 32. Install backplane circuit board (Sol-BPB). The photograph in Figure 6-7 on Page VI-17 shows the backplane board installed.

- \* ( ) Position Sol-BPB with 100-pin male edge connector down and the five female edge connectors facing the card guides. The board should rest against the front face of the right angle brackets as shown in Figure 6-6. Adjust position of Sol-PC as needed so that you can plug the Sol-BPB edge connector into J11 on the Sol-PC.
- \* ( ) Align holes on left and right ends of Sol-BPB with those in right angle brackets.

(Step 32 continued on Page VI-17)



\* #6 x  $\frac{1}{4}$  sheet metal screw

+ 4-40 x  $\frac{5}{8}$  binder or pan head screw,  
#4 lockwasher and 4-40 hex nut

Figure 6-6. Backplane board (Sol-BPB) installation.

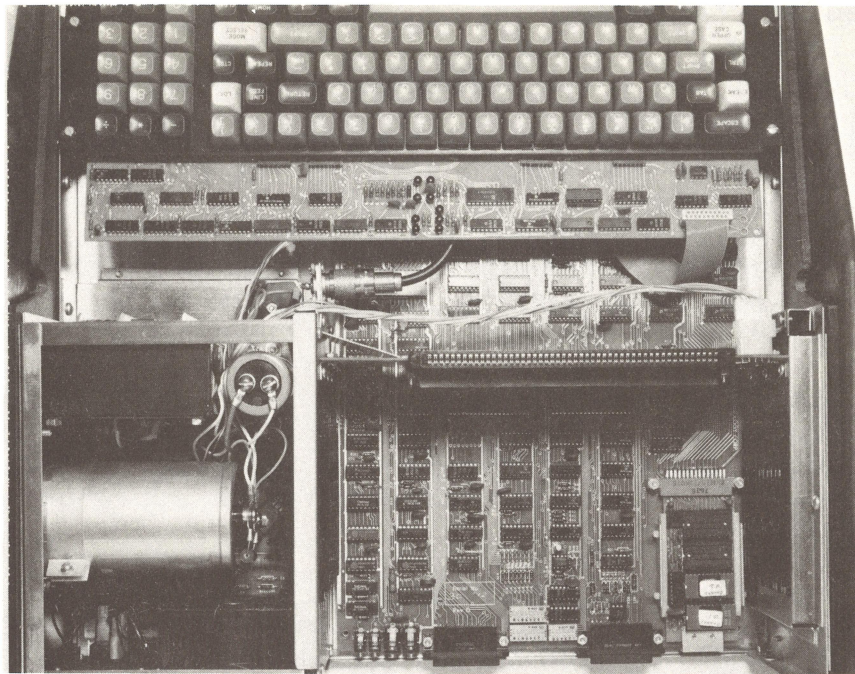


Figure 6-7. Backplane board (Sol-BPB) installation. Rear of Sol is at bottom and Sol-BPB is to right of power supply subchassis in line with C8 and transformer.

(Step 32 continued)

- \*( ) Attach Sol-BPB to brackets with three 4-40 x 5/16 binder or pan head screws, #4 lockwashers and 4-40 hex nuts on each side. Insert screws from the back side of bracket through Sol-BPB, place lockwasher on each screw and secure each with nut.
- \*( ) Step 33. Install left and right gusset brackets as shown in Figure 6-6 on Page VI-16.
- \*( ) Fit narrower gusset bracket on left side so that its flanges are flat against the expansion chassis side wall and the backplane board. (You may have to bend the flange slightly to obtain a proper fit.)
- \*( ) Attach bracket to expansion chassis side wall with the three #6 x  $\frac{1}{4}$  sheet metal screws you used in Step 19 to prethread the holes.

**\*\*See WARNING on Page VI-18.\*\***

(Step 33 continued on Page VI-18.)

(Step 33 continued.)

WARNING

IT IS QUITE EASY TO SCRATCH OR CUT YOUR HAND ON THE SOLDER SIDE OF THE BACKPLANE BOARD WHEN DRIVING THESE SCREWS. PLACE A SUITABLE PROTECTIVE BARRIER, SUCH AS CARDBOARD, AGAINST SOLDER SIDE OF BACKPLANE BOARD DURING INSTALLATION TO PREVENT SUCH INJURY.

- \* ( ) Attach bracket to backplane board with three 4-40 x 5/8 binder or pan head screws, #4 lockwashers and 4-40 hex nuts. Insert screws from front side of bracket through Sol-BPB, place lockwasher on each screw and secure each with nut.
- \* ( ) Install wider gusset bracket on right side in the same manner as you did the left bracket. THE PRECEDING WARNING ALSO APPLIES TO INSTALLING THIS BRACKET.
- \* ( ) Step 34. Connect Sol-20 DC power cable from power supply subchassis to the Sol-BPB power cable you installed in Step 4.
- ( ) Step 35. Check that Sol-PC is in optimum position and tighten the eight screws holding the Sol-PC to the expansion-main chassis assembly. (See Step 27.)
- ( ) Step 36. Connect Sol-PC power cable (4-wire) to J10 on Sol-PC. CAUTION: Make sure cable connector mates exactly with J10; that is, pin 1 to pin 1, pin 2 to pin 2, etc. Any other mating relationship will damage the IC's on the Sol-PC. (Refer to Step 15 in Section III.)
- ( ) Step 37. Position keyboard (Sol-KBD) near its mounting brackets and connect 20-conductor ribbon cable supplied with Sol keyboard kit between J1 on keyboard and J3 on Sol-PC. With the cable connected properly, the cable will run away from the keys from J1 on the keyboard, and towards the keys from J3 on Sol-PC.
- ( ) Step 38. Attach keyboard to keyboard brackets with two 6-32 x 1/2 binder or pan head screws and #6 lockwashers on each side. Place washer on each screw and drive screws loosely into threaded holes in brackets.

- ( ) Step 39. If your kit does not include the 15-key numeric pad, install the plastic insert supplied with your Sol keyboard kit to the keyboard cover. Attach it on the right end and to the bottom of the cover with masking tape.
- ( ) Step 40. Remove protective cover from one side of Plexiglass strip and attach "Sol Terminal Computer" trim plate to Plexiglass with small pieces of transparent tape. Place trim plate with printed side against Plexiglass.
- ( ) Step 41. Remove protective cover from other side of Plexiglass and slide it into the channel above the keyboard cutout.

NOTE

A hole is provided in the sheet metal behind the trim plate. This may be used for a "power on" indicator light if desired.

- ( ) Step 42. Install keyboard cover. Hook front of cover under front edge of main chassis and lower it over the keyboard. (A slight adjustment of the keyboard position may be needed to obtain a proper fit.)

Position keyboard within cutout in cover if needed and tighten keyboard mounting screws.

- ( ) Step 43. Install top cover.
  - ( ) Be sure power cord is not plugged into 110 V ac outlet and disconnect cord from fan closure plate receptacle.
  - ( ) Remove fuse holder cap and fuse.

CAUTION

NEVER REMOVE OR INSTALL FUSE WITH POWER ON.

- ( ) Hook top cover over back edge of keyboard cover and lower it down into place over the rear of the main chassis. Install the two thumb screws (one at the lower left corner and the other to the right of the fan closure plate coaxial connector) to attach cover to rear of main chassis.
- ( ) Step 44. Re-install fuse and plug power cord into receptacle. BE SURE POWER CORD IS NOT PLUGGED INTO 110 V ac OUTLET.

See CAUTION on Page VI-20.

(Step 44 continued.)

CAUTION

NEVER REMOVE OR INSTALL FUSE WITH POWER ON.

- ( ) Step 45. Remove backing from connector identification label and affix it to rear of top cover. Position label just above Sol-PC connector opening in cover so that "J9" is aligned with left most (as viewed from rear of Sol) subminiature phone jack and "J1" is aligned with right most 25-pin female connector.
- ( ) Step 46. Remove backing from serial number label and affix it to rear of top cover. Position label to right (as viewed from rear of Sol) of fan opening in cover.
- ( ) Step 47. Affix self-stick protective pads to bottom of Sol as shown in Figure 6-8.

You have now completed assembly of your Sol Terminal Computer<sup>TM</sup>. It is ready for use as a stand-alone computer or CRT terminal. Congratulations on a job well done.

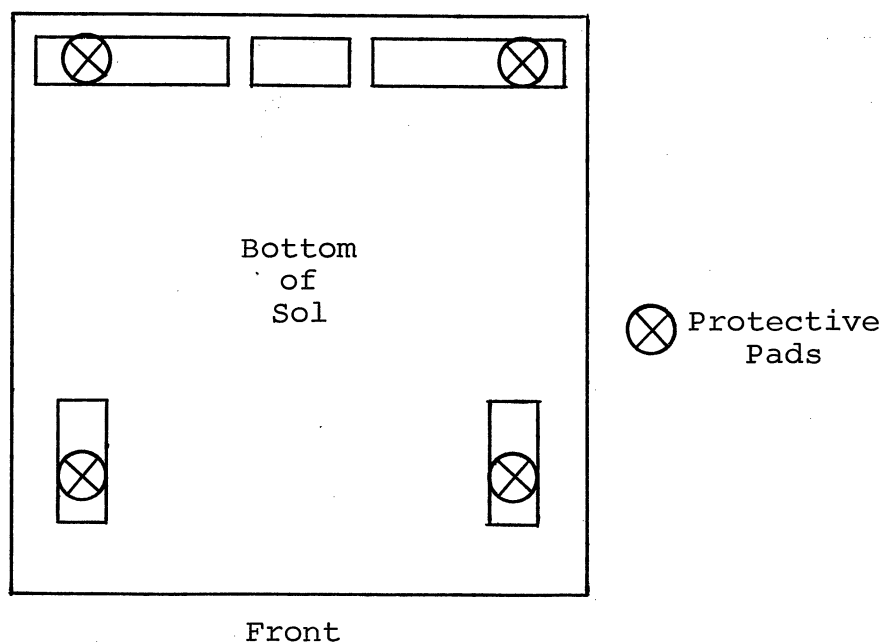


Figure 6-8. Protective foot pad installation.



## PROCESSOR TECHNOLOGY CORPORATION

## SECTION VII, Sol OPERATING PROCEDURES

## CONTENTS

<u>PARAGRAPH</u>	<u>TITLE</u>	<u>PAGE</u>
7.1	INTRODUCTION . . . . .	VII-1
7.2	THE OPERATING CONTROLS . . . . .	VII-1
7.3	BASIC OPERATING MODES . . . . .	VII-1
7.3.1	Command Mode . . . . .	VII-1
7.3.2	Terminal Mode . . . . .	VII-4
7.4	GETTING ACQUAINTED WITH Sol . . . . .	VII-4
7.4.1	Monitor and Cassette Recorder Connections . . . . .	VII-4
7.4.2	Terminal Mode Operation . . . . .	VII-5
7.4.3	Command Mode Operation . . . . .	VII-10
7.5	OPERATING CONTROLS IN DEPTH . . . . .	VII-12
7.5.1	ON-OFF Switch . . . . .	VII-13
7.5.2	Restart (RST) Switch, S1-1 . . . . .	VII-13
7.5.3	Control Character Blanking (BLANK) Switch, S1-3 . . . . .	VII-13
7.5.4	Video Display (POLARITY) Switch, S1-4 . . . . .	VII-13
7.5.5	Cursor Selection (BLINK, SOLID) Switches, S1-5 & 6 . . . . .	VII-13
7.5.6	Sense (SSWØ - 7) Switches, S2-1 through S2-8 . . . . .	VII-13
7.5.7	Baud Rate Switches, S3-1 through S3-8 . . . . .	VII-14
7.5.8	Parity (PS, PI) Switches, S4-1 & 5 . . . . .	VII-14
7.5.9	Data Word Length (WLS-1 & 2) Switches, S4-2 & 3 . . . . .	VII-14
7.5.10	Stop Bit Selection (SBS) Switch, S4-4 . . . . .	VII-14
7.5.11	Full/Half Duplex (F/H) Switch, S4-6 . . . . .	VII-15
7.5.12	Keyboard . . . . .	VII-16
7.6	THE KEYBOARD, GENERAL DESCRIPTION . . . . .	VII-16
7.6.1	Operating Features . . . . .	VII-16
7.6.2	Keyboard Indicators . . . . .	VII-16
7.7	INDIVIDUAL KEY DESCRIPTIONS . . . . .	VII-17
7.7.1	Alphanumeric-Punctuation-Symbol Keys . . . . .	VII-17
7.7.2	Space Bar . . . . .	VII-21
7.7.3	Arithmetic Pad Keys . . . . .	VII-21
7.7.4	ESCAPE Key . . . . .	VII-22
7.7.5	BREAK Key . . . . .	VII-22
7.7.6	TAB Key . . . . .	VII-22
7.7.7	Control (CTRL) Key . . . . .	VII-22
7.7.8	SHIFT Key and SHIFT LOCK Key/Indicator . . . . .	VII-22
7.7.9	UPPER CASE Key/Indicator . . . . .	VII-22
7.7.10	LOCAL Key/Indicator . . . . .	VII-24

(Continued on next page.)



## PROCESSOR TECHNOLOGY CORPORATION

## SECTION VII, Sol OPERATING PROCEDURES

## CONTENTS

<u>PARAGRAPH</u>	<u>TITLE</u>	<u>PAGE</u>
7.7	INDIVIDUAL KEY DESCRIPTIONS (Continued)	
7.7.11	RETURN Key . . . . .	VII-24
7.7.12	LINE FEED Key . . . . .	VII-24
7.7.13	LOAD Key . . . . .	VII-24
7.7.14	REPEAT Key . . . . .	VII-24
7.7.15	MODE SELECT Key . . . . .	VII-24
7.7.16	CLEAR Key . . . . .	VII-24
7.7.17	Cursor Control (HOME CURSOR and Arrows) Keys	VII-25
7.8	BASIC OPERATIONS . . . . .	VII-25
7.8.1	Switching From Terminal To Command Mode . .	VII-25
7.8.2	Switching From Command To Terminal Mode . .	VII-25
7.8.3	Entering Commands In The Command Mode . . .	VII-25
7.8.4	Keyboard Restart . . . . .	VII-26
7.9	Sol-PERIPHERAL INTERFACING . . . . .	VII-26
7.9.1	Audio Cassette Recorders . . . . .	VII-26
	<u>Write Operations</u> . . . . .	VII-28
	<u>Read Operations</u> . . . . .	VII-28
7.9.2	Serial Data Interface (SDI) . . . . .	VII-29
7.9.3	Parallel Data Interface (PDI) . . . . .	VII-30
7.10	CHANGING THE FUSE . . . . .	VII-32

Cassette Recorders for use with Sol

Not all audio cassette recorders are suited for data storage use with the Sol. The following models have been tested and approved by Processor Technology:

1. Panasonic RQ-413AS
2. Realistic CTR-21

Some users have reported unsuccessful results with the Panasonic RQ-309 and the J. C. Penny Catalog #851-0018. If you should wish to select a different model, the following features, included on the models above, are necessary:

1. An AUX input. Although the Sol can be jumpered for low level Microphone level input, the procedure is no longer recommended.
2. A digital counter. The counter is necessary in locating programs on the cassette.
3. A tone control. The existence of a tone control is one indication of high quality electronics.

Even though a recorder has the three features, there is no guarantee that it will work properly for the purpose. Recorders vary greatly in the quality of their electronics. If possible, test the recorder with a long file before purchasing it, in both record (SAVE) and playback (GET or XEQ) mode. If the recorder is not working properly, either you will get an error message, or you will find differences between what was recorded and what was played back.

Observe the following pointers for best results:

1. Keep the recorder at least a foot away from the Sol, or other equipment which can generate magnetic fields. The recorder can pick up hum which may generate errors.
2. Keep the tape heads cleaned and demagnetized in accordance with the manufacturer's instructions.
3. Use high quality brand-name tape. Cheap tape can wear down the tape heads and give erratic results.
4. Bulk erase tapes before using.
5. Keep the cassettes in their protective plastic covers, in a cool place, when not in use. Cassettes are vulnerable to dirt, high temperatures, liquids, and physical abuse.
6. Set the tone control at midrange, and set the volume control about 2/3 full volume. The Sol has an automatic gain control circuit which compensates for a wide range of levels, but operation in the middle of this range will



give the most reliable results. Experiment to find the best setting of volume and tone controls.

7. On some cassette recorders, the microphone can be live while recording through the AUX input. Deactivate the mike in accordance with the manufacturer's instructions. In some cases this can be done by inserting a dummy plug into the microphone jack.
8. During recording, some recorders present the signal being recorded at the monitor or earphone output. In a system with two cassette recorders this could cause problems if an attempt was made to read from one recorder while the other was writing. Since both recorders share the same audio lines, the monitor output of the recorder which was recording could interfere with the signal being read from the other recorder.
9. If you record more than one file on a tape side, SAVE a special file, which could be named END, to let you know when you have played past the files of interest. After recording the last file on a side, rewind the tape, set the digital counter to zero, and issue a CATalog command (see SOLOS/CUTER User's Manual). As each file header is displayed, make a note of the reading on the digital counter, the exact name of the file, load address, and file length. Mark the cassette with this information to make file retrieval much easier.

If you experience a read error, use the following procedure to isolate the problem:

1. Check for proper settings, and make sure you have followed the pointers above.
2. Check cables for intermittent connections and shorts.
3. Note the exact reading of the digital counter at the time of the error.
4. Rewind the tape and try to read the same part of the tape again. If the tape reads without errors this time, the error was not recorded on the tape. If there is a read error at the same point, then the error is recorded on the tape.
5. Rewind the tape and record a file on the same part of the tape. Read the file. If the tape reads without errors, then the original read error was generated during the recording process. If there is still a read error at the same point, then the cassette itself is faulty.

## 7.1 INTRODUCTION

Information in this section will help you to become familiar with the operation of your Sol Terminal Computer<sup>TM</sup>. Following brief explanations of the operating controls and the two basic operating modes, you will put your Sol through some simple operations. This should sufficiently acquaint you with the keyboard and control switches so that you will feel at ease with your Sol. In addition, you will have performed functional tests of all Sol sections except the parallel data interface.

Detailed descriptions of the control switches are also provided to allow you to gain greater proficiency in their use. For the same reason, individual keyboard key descriptions are also given. They are intended to be used along with the BASIC/5 and SOLOS Users' Manuals (or if applicable the CONSOL description in Section IX of this manual).

The balance of this section supplies instructions for 1) connecting typical peripheral devices to the serial and parallel data interfaces (J1 and J2), 2) using audio cassette recorders, and 3) changing the fuse.

## 7.2 THE OPERATING CONTROLS

Sol operating controls are identified and their functions briefly defined in Table 7-1 on Page VII-2. Unless noted otherwise, the location of each control is shown on the Sol-PC assembly drawing in Section X, Page X-3.

## 7.3 BASIC OPERATING MODES

### 7.3.1 Command Mode

In this mode Sol operates as a stand alone computer under control of the program (software) contained in the personality module and additional software that is stored in the Sol, stored either in a read only memory (ROM) that is plugged into the computer or the Sol random access memory (RAM). (For a description of the CONSOL and SOLOS Personality Modules, refer to Section IX in this manual and the SOLOS Users' Manual respectively.)

With the SOLOS Personality Module installed, the computer is in the command mode when power is applied to the Sol. Command mode is a sort of "home base" from which excursions may be made into other programs. An analysis of three levels of programs will make the concept of command mode more understandable.

At the lowest level of software are the instructions which the 8080 CPU (central processing unit), the brains of the computer,

Table 7-1. Sol Operating Controls and Their Functions.

CONTROL	FUNCTION
ON-OFF Switch (See Figure 7-1)	Connects and disconnects primary power to Sol.
RST (Restart) Switch, S1-1	Permits manual restart of Sol without turning power off. (Useful for test purposes.)
BLANK Switch, S1-3	Determines if control characters are displayed or not.
POLARITY Switch, S1-4	Selects normal (white characters on black background) or reverse video display.
BLINK-SOLID Switches, S1-5 & 6	Selects blinking, nonblinking or no cursor.
SSWØ - 7 S2-1 through 8	Permits direct data entry to processor.
BAUD RATE Switches, S3-1 through 8	Sets operating speed of serial data interface (SDI).
PS & PI Switches S4-1 & 5	Selects no parity, even parity or odd parity for SDI.
WLS-1 & 2 Switches, S4-2 & 3	Selects number of data bits in transmitted word for SDI.
SBS Switch, S4-4	Determines number of stop bits in transmitted word for SDI.
F/H Switch, S4-6	Selects half or full duplex operation for SDI.
Keyboard (See Figure 7-4)	Data entry, mode selection, command input and cursor control.

can understand and run. All programs must ultimately be reduced to this basic level to be operated on by the computer. In the case of the 8080 microprocessor, the program is in an "object code" or "machine language", since the "machine" or 8080 CPU understands it. The SOLOS program contained in the personality module is stored in this machine language form, and the computer can therefore run directly from this program. Since the SOLOS program is contained in permanent ROM which is plugged directly into the computer, the SOLOS program is always available, and is automatically selected whenever the power switch of the Sol is turned on. There is also provision for returning at all times to the command mode of SOLOS. From the command mode other programs may be brought in for various operations or stored on cassette tape. The contents of the computer's memory may be displayed or changed. The command mode also performs "housekeeping" functions such as setting the rate at which data is read from tape, or the rate at which characters are displayed on the video monitor.



The command mode allows the introduction of the second level of software. This level includes higher-level language programs such as BASIC/5 or FOCAL in which complex application programs may be more easily written. These are called higher level languages because they permit the user to write programs in a form much closer to human languages such as English. However, programs written in these languages must be translated into the more basic machine language before they can be run. Besides higher level languages, this second level of software includes programs such as the TREK 80 and GAMEPAC video games and the ALS-8 program (a software package used for developing programs), all of which are offered by Processor Technology Corporation. Through the facilities of the command mode, these second level programs are transferred (loaded) into memory from cassette tape or other storage media, and then "executed" (used). These programs may also exist in ROM or EPROM (erasable programmable ROM) memory which is plugged into the computer to make them instantly available like the SOLOS program. All first and second level programs are stored in the computer as binary object code.

Let us illustrate the concept of the second level of programs with an example, BASIC/5. Using the "XEQ" command available in the SOLOS command mode, we load the BASIC/5 program into the computer's memory from cassette tape. With this command BASIC/5 is ready for use as soon as the tape has stopped moving. The control of the computer is now taken over by the BASIC/5 program now in memory, and SOLOS is no longer in command. All the features of BASIC/5 language are now available to us, with a new set of commands and rules. Since the CPU of the computer only understands the machine language of the first level of software, the BASIC/5 program must translate the commands and data we enter to this lower level. BASIC/5 does this as we go. While we are using BASIC/5, we still have access to some of the commands and features of SOLOS, although they may have a modified form while we are in BASIC/5. We will load and use BASIC/5 later in this section.

The third level of software consists of programs written using the higher order languages of the second level programs. A program written in BASIC/5 is on this third level. This program only makes sense to the computer while the computer has BASIC/5 in memory and control has been transferred to the BASIC/5 program. Third level programs written in any high level language are often called "applications programs" since they are usually written in order to fit a specific application need.

The ALS-8 Program Development System is another second level program. A program to be developed within ALS-8 would then be a third-level application program. The ALS-8 also includes an Assembler which takes a program written on the third level in "assembly" language, and translates it to object code which the computer can run. The object code version then resides in memory and can be run in another operation. For a further discussion of types of software see the article "Your Personal Genie" in Appendix VIII of this manual.

### 7.3.2 Terminal Mode

Sol operates as a CRT terminal in this mode, capable of sending keyboard data to an output port and displaying data received at the serial input port on an external video monitor via the Sol video display circuitry. When Sol is "hard-wired" to another computer or connected to a modem, the terminal mode is used for data entry, data retrieval, inquiry/response and monitoring and control applications.

Capabilities in the terminal mode depend on the personality module used. Both CONSOL and SOLOS Personality Modules permit operation as a CRT terminal. CONSOL 1) initializes Sol in the terminal mode whenever you turn the power on or initiate a system reset, 2) sends keyboard data to the serial data interface (SDI) only, and 3) provides simple stand-alone computer capabilities. SOLOS, on the other hand, 1) enters the terminal mode when given the "TERM" (terminal) command, 2) sends keyboard data to any output port available with the "SET O" (set out) command, and 3) duplicates CONSOL functions while providing additional capabilities.

## 7.4 GETTING ACQUAINTED WITH Sol

One of the best ways to get acquainted with your Sol is to use it. After connecting a cassette recorder and video monitor to your Sol, you will operate the system in the terminal mode to become familiar with the keyboard and the functions of the video display switches. You will then switch to the command mode and perform some of the basic computer operations.

### 7.4.1 Monitor and Cassette Recorder Connections

The basic Sol system consists of the Sol, a video monitor for display (e.g., the Processor Technology PT-872 TV-Video Monitor by Panasonic) and a cassette recorder for external storage (e.g., the Panasonic Model RQ-413S).

To connect these three system components, you will need the following cables:

Audio In & Out Cables--two cables of shielded wire fitted with miniature phone plugs at both ends.

Motor 1 Cable--one cable pair, such as speaker wire, fitted with subminiature phone plugs at both ends. (An identical cable for Motor 2 is needed if you use two recorders.)

Video Cable--one RG59/U coaxial cable fitted with a PL259 UHF male connector on one end and a monitor-compatible connector on the other.

Connect the basic Sol system as follows (refer to Figure 7-1 on Page VII-6):

- ( ) Step 1. Remove top and keyboard covers from Sol.
- ( ) Step 2. Plug one end of Audio In Cable into Audio IN jack (J7) on Sol rear panel, and plug other end into MONITOR or EARPHONE jack on recorder.
- ( ) Step 3. Plug one end of Audio Out Cable into Audio OUT jack (J6) on Sol rear panel, and plug other end into AUXILIARY or MICROPHONE jack on recorder. (The AUXILIARY input is preferred and recommended over the MICROPHONE input.)

NOTE

If your recorder has only a microphone jack, remove the I-to-J jumper installed in Step 69 in Section III and install a jumper between I and H.

- ( ) Step 4. Plug one end of Motor 1 Cable into Motor 1 jack (J8) on Sol rear panel, and plug other end into REMOTE jack on recorder.
- ( ) Step 5. Connect PL259 UHF connector on Video Cable to video output connector on Sol rear panel, and connect other end to video monitor input connector.
- ( ) Step 6. Make sure monitor, recorder and Sol power switches are in their OFF position. Then connect AC power cord to AC receptacle on Sol rear panel and connect Sol, monitor and recorder to appropriate power source.

#### 7.4.2 Terminal Mode Operation

The following procedure assumes your Sol is equipped with a SOLOS personality module.

- ( ) Step 7. Set Sol control switches as follows (see Figure 7-2 on Page VII-7):

RST Switch (S1-1): OFF

S1-2 (spare): OFF

BLANK Switch (S1-3): OFF (display control characters)

POLARITY Switch (S1-4): OFF (reverse video display)

BLINK Switch (S1-5): OFF (solid cursor)

SOLID Switch (S1-6): ON (solid cursor)

(Step 7 continued on Page VII-7.)

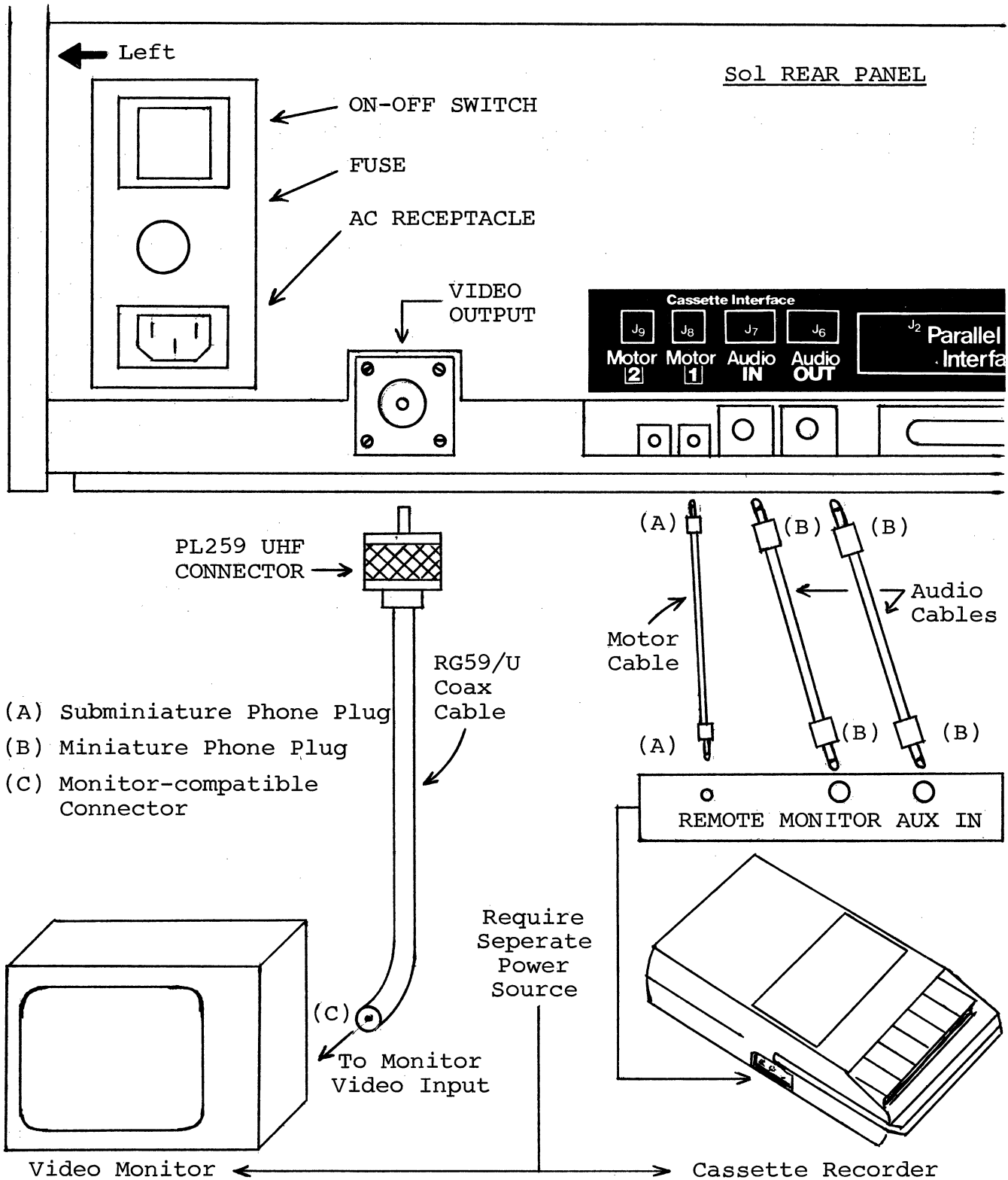


Figure 7-1. Connecting the basic Sol system.

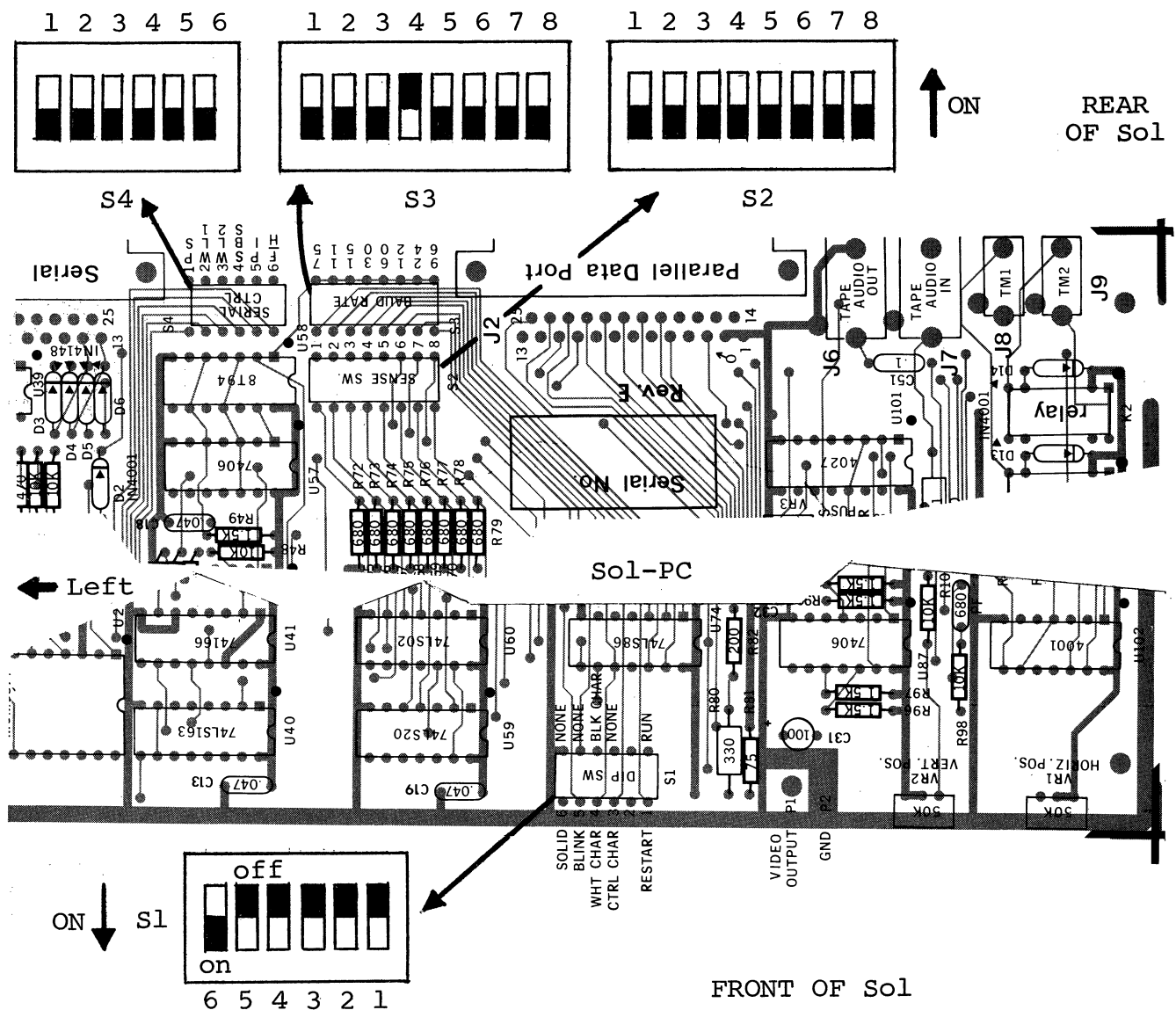


Figure 7-2. Sol control switch settings for terminal mode.

(Step 7 continued.)

SSW Switches (S2-1 - 8): OFF

BAUD RATE Switches (S3-1 - 8): S3-4 ON, all others OFF  
(300 Baud)SDI Switches (S4-1 - 6): OFF (selects full duplex operation,  
8 data bits, 2 stop bits and no  
parity)

- ( ) Step 8. Turn Sol and monitor on.
- ( ) Step 9. If the monitor display raster is out of sync (black horizontal bar moves slowly down screen, numerous black lines cut across raster, or both), adjust monitor vertical and horizontal hold controls for a stable raster.
- ( ) Step 10. You should see a prompt character followed by the cursor ( > | ) in the upper left corner of the screen. If you don't, adjust VR1 and VR2 (see Figure 7-3) to move the prompt character and cursor onto the screen. (With CONSOL, only the cursor will appear on the screen.)

**NOTE**

Use VR1 (horizontal position) and VR2 (vertical position) to center the display page (16 lines, 64 characters/line) on the screen.

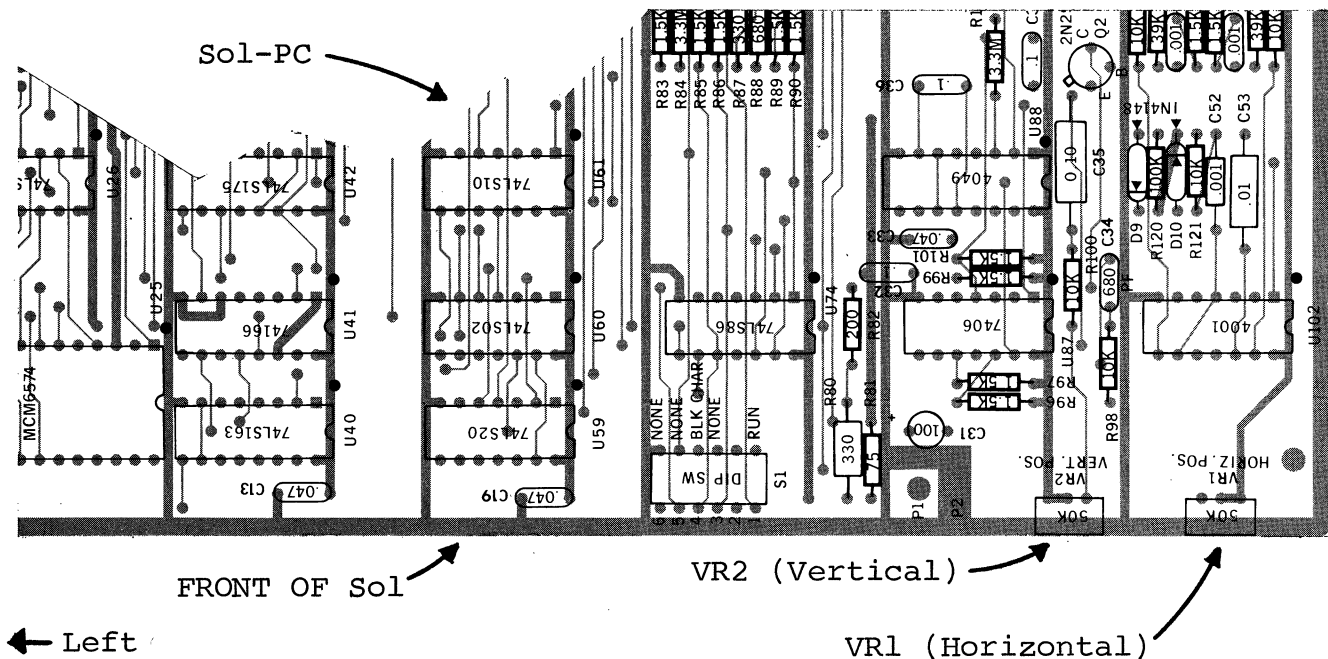


Figure 7-3. Location of positioning adjustments, VR1 and VR2.

- ( ) Step 11. Enter terminal mode by 1) pressing UPPER CASE key to turn the indicator light on (Alphabetic characters are now entered as upper case, regardless of SHIFT key status, but dual character keys do respond to SHIFT key.), 2) typing TERM and 3) pressing RETURN key. (If your Sol is equipped with CONSOL, it entered terminal mode when you turned the Sol on.) "TERM" will appear on the screen as you type, and the cursor will disappear when you press the RETURN key.



## PROCESSOR TECHNOLOGY CORPORATION

## Sol OPERATING PROCEDURES

## SECTION VII

NOTE: All commands must be given in upper case characters in order to be recognized, and the RETURN key must be pressed after a command so that SOLOS can execute the command (MODE SELECT excepted).

- ( ) Step 12. Set for local operation by pressing LOCAL key to turn indicator light on. Set for lower case operation by pressing UPPER CASE again (indicator light out).
- ( ) Step 13. Press each of the alphanumeric, punctuation and symbol keys. As each is pressed, the lower case character in the UNSHIFTED column of Table 7-4 should appear on the screen. Read Section 7.7 on page VII-17 to become familiar with Table 7-4.

NOTE: If the MODE SELECT key is pressed, SOLOS will return to the command mode and display a prompt character followed by the cursor. In this case return to terminal mode by typing "TERM" in upper case letters, followed by a carriage return.

- ( ) Step 14. Press SHIFT LOCK key to return keyboard to shifted operation (indicator light will go out) and repeat Step 13. Each corresponding upper case character should appear from the SHIFTED column of Table 7-4.
- ( ) Step 15. Use the control sequences given in Table 7-4 on Page VII-18 to generate the indicated control characters. Control characters are generated by pressing the CTRL (control) key and, while holding it depressed, pressing the desired key given in the first column of the table. As the table shows in the last two columns, the symbol generated by a control sequence depends on whether a 6574 or 6575 character generator (U25) is installed in your Sol. Two examples follow:

<u>CONTROL SEQUENCE</u>	<u>6574 SYMBOL</u>	<u>6575 SYMBOL</u>
CTRL and I	→	H <sub>T</sub>
CTRL and 5 or %	⊗	E <sub>Q</sub>

- ( ) Step 16. Change video display polarity by setting POLARITY Switch (S1-4) to ON and observe the effect on the display. It should change from black characters on a white background to white characters on a black background.
- ( ) Step 17. Switch from non-blinking cursor to a blinking cursor by setting SOLID Switch (S1-6) to OFF and BLINK Switch (S1-5) to ON in that order. You should see a rectangular solid cursor that blinks on and off approximately two times per second. Never put S1-5 and S1-6 ON at the same time.
- ( ) Step 18. Blank control characters by setting BLANK Switch (S1-3) to ON. Any control characters generated (refer to Step 15) should not appear on the screen.

Up to this point, keyboard data has been processed by the CPU, transmitted out through the serial channel output, looped back

to the serial channel input and then displayed on the video monitor. You have consequently just "tested" the CPU, serial channel and display section functions in your Sol.

#### 7.4.3 Command Mode Operation

The following operations assume your Sol is equipped with a SOLOS personality module.

Using the Cassette Recorder. The following procedure for loading a program from cassette tape into Sol memory provides a good example of how to use an audio cassette recorder with Sol. In this example you will use the BASIC/5 cassette supplied with your Sol.

- ( ) Step 19. Set POLARITY (Sl-4) and BLANK (Sl-3) Switches as desired.
- ( ) Step 20. Replace top and keyboard covers.
- ( ) Step 21. Load BASIC/5 cassette in recorder. If required, fully rewind tape. (This can be done by disconnecting the REMOTE plug from the recorder and using the REWIND control on the recorder.) After rewinding, reconnect REMOTE plug.
- ( ) Step 22. Set the following recorder controls and indicator, if so equipped, as indicated:

Transport: press STOP control

Volume: midrange

Tone: top of range (maximum treble)

Tape Counter: Ø

- ( ) Step 23. Press PLAY control on recorder. The tape should not move. If it does, there is a malfunction in the remote control circuitry or cabling. (With the Sol off, there should be no continuity between the REMOTE plug contacts.)

#### NOTE

The tape head must be clean to reliably read a tape or write on tape.

- ( ) Step 24. If needed, press MODE SELECT key on Sol to enter command mode. (Remember SOLOS initializes in the command mode while CONSOL initializes in the terminal mode whenever Sol is turned on.) You should see a prompt character followed by the cursor ( >| ) on the left of the screen.

PROCESSOR TECHNOLOGY CORPORATION

Sol OPERATING PROCEDURES

SECTION VII

- ( ) Step 25. Type the XEQ command as follows:

XEQ BASIC

- ( ) Step 26. Press the RETURN key on Sol. The cursor should disappear and the tape should advance. The display should not change otherwise. NOTE: With certain cassette recorders or cassettes there may be a misreading of the tape when the splice joining the leader to the tape passes the tape head. In this case an ERROR message will appear and the tape will stop. To resume tape "loading", retype the XEQ BASIC command. If further difficulty is encountered, try different cassette recorder volume settings until a reliable setting is found.

- ( ) Step 27. If the tape has loaded successfully, in approximately two minutes BASIC/5 will display five lines of text ending with:

SOL BASIC 5

READY

- ( ) Step 28. BASIC/5 is now ready for use. Refer to your BASIC/5 User's Manual. Become familiar with both BASIC/5 and the Sol keyboard. Try some exercises in BASIC/5.

Dump Operation. The dump operation displays memory data in hexadecimal on the video monitor. It can also be used with the appropriate SET command to output memory data to a hard-copy device (e.g., a printer). As an example, dump the first part of the SOLOS personality module (C000 through C0E0) as follows:

- ( ) Step 29. Set UPPER CASE key so that the indicator is on. If you are still in BASIC/5, type the BASIC/5 command "BYE" at the beginning of a command line to re-enter SOLOS command mode. BASIC/5 remains in memory and may be returned to by typing a command line: "EXEC 0".

- ( ) Step 30. Type the DUMP command as follows:

DUMP C000 C0E0

- ( ) Step 31. Press RETURN key. Lines of 16 bytes of hexadecimal data will scroll (move) rapidly up the screen until the last address (C0E0) is displayed. At this point the display will stop scrolling.

Enter Operation. The enter operation is used to enter hexadecimal data from the keyboard into available Sol memory. As an example, enter 16 bytes of data, starting at address C900 and ending at address C90F, as follows:

- ( ) Step 32. Type the ENTER command as follows:

ENTER C900

- ( ) Step 33. Press RETURN key. The monitor should display a colon (:) prompt character at the start of the next line.

- ( ) Step 34. Type the following data:

11 22 33 44 55 66 77 88 99 00 AA BB CC DD EE FF/

#### NOTE

The slash (/) terminates the enter function.

- ( ) Step 35. If you made a mistake in typing the above line of data, refer to Paragraph 7.8.3 on Page VII-25. If you made no mistakes, press RETURN key.

The data entered in Step 34 now resides in locations C900 through C90F in the Sol memory.

- ( ) Step 36. To verify that the data did indeed enter Sol memory, simply give your Sol this DUMP command:

DUMP C900 C90F

Then press RETURN key. The line of data you entered in Step 35 should be displayed on the monitor screen, preceded by the starting address.

- ( ) Step 37. Using your SOLOS User's Manual, experiment with the other commands until you feel at home with your Sol.

The preceding command mode operations used the CPU, personality module, audio cassette interface (ACI) and the Sol RAM. You have consequently just tested the functions of these sections.

## 7.5 OPERATING CONTROLS IN DEPTH

Unless indicated otherwise, the location of the controls described in this paragraph are shown on the Sol-PC assembly drawing in Section X, Page X-3.

## 7.5.1 ON-OFF Switch (See Figure 7-1 on page VII-6.)

Push this switch in to turn your Sol on. In the ON position the switch remains locked in its "in" position. To turn your Sol off, push the switch again. This releases the locking mechanism, and the switch will return to its OFF ("out") position.

## 7.5.2 Restart (RST) Switch, Sl-1

This switch permits you to restart your Sol without turning the power off. You should normally leave it in its OFF, or run, position. Set it to ON and then OFF to initialize the Sol circuitry and reset the CPU program counter to zero. (A manual restart with this switch performs the same function as turning the power on or pressing a keyboard generated restart: UPPER CASE key with REPEAT key.)

## 7.5.3 Control Character Blanking (BLANK) Switch, Sl-3

Set this switch to its ON position if you do not want control characters (see Table 7-4 on Page VII-18) to be displayed on the screen. In the OFF position, control characters are displayed.

## 7.5.4 Video Display (POLARITY) Switch, Sl-4

If you want a normal video display (white characters on a black background), set this switch to its ON position. In the OFF position, black characters will be displayed on a white background (reverse video display).

## 7.5.5 Cursor Selection (BLINK, SOLID) Switches, Sl-5 &amp; 6

CAUTION

DO NOT SET Sl-5 AND Sl-6 TO THEIR ON POSITIONS AT THE SAME TIME. TO DO SO MAY DAMAGE YOUR Sol.

If you want the cursor to blink, set Sl-6 to OFF and Sl-5 to ON. The cursor will blink on and off about two times per second.

Set Sl-5 to OFF and Sl-6 to ON if you want a non-blinking (solid) cursor.

With both Sl-5 and Sl-6 in their OFF positions, there will be no cursor display.

## 7.5.6 Sense (SSWØ - 7) Switches, S2-1 through S2-8

These eight switches are normally left in the OFF position. They are used to manually enter data into the CPU. (They serve the same function as the front panel sense switches on the Altair 8800 and IMSAI 8080.)

S2-1 is the least significant data bit (DIO0) and S2-8 is the most significant data bit (DIO7). To pull a DIO bit low (when the program tests SSW0 - 7), set the switch associated with the bit to ON. An open (OFF) switch pulls the associated DIO bit high when the program tests SSW0 - 7.

NOTE

The configuration of SSW0 - 7 is tested by the CPU only when it executes an input port FF instruction. Otherwise, the Sense Switches have no bearing on Sol operation.

7.5.7 Baud Rate Switches, S3-1 through S3-8

The setting of the Baud Rate Switches determines the operating speed of the Serial Data Interface (SDI). Assuming you have not installed any of the K, L and M jumper options, you can select any one of eight Baud rates. Table 7-2 on page VII-15 defines Baud rate as a function of S3-1 through S3-8.

CAUTION

DO NOT SET MORE THAN ONE S3 SWITCH TO THE ON POSITION AT THE SAME TIME. TO DO SO CAN DAMAGE YOUR Sol.

7.5.8 Parity (PS, PI) Switches, S4-1 & 5

With these two switches you can select no parity, parity, even parity or odd parity for data handled through the SDI (J1).

Set S4-5 (PI) to its ON position if you want a parity bit. When OFF, there will be no parity bit. (A stop bit immediately follows the data if no parity bit is selected.)

S4-1 (PS) selects even or odd parity if S4-5 is ON. It otherwise has no affect. For even parity, set S4-1 to ON. Set S4-1 OFF for odd parity.

7.5.9 Data Word Length (WLS-1 & 2) Switches, S4-2 & 3

Use these two switches to select the number of bits, excluding parity, in the transmitted word for the SDI. You have a choice of 5, 6, 7 or 8 bits. Table 7-3 defines word length as a function of S4-2 and S4-3.

7.5.10 Stop Bit Selection (SBS) Switch, S4-4

Set this switch to ON if you want one stop bit transmitted out of the SDI. In the OFF position, two stop bits are transmitted unless you have selected a five bit word length. In that case 1.5 stop bits are transmitted.



Table 7-2. Baud Rate Selection With Switch S3.

BAUD RATE	SWITCH S3 CONFIGURATION*
75	S3-1 ON, all others OFF
110**	S3-2 ON, all others OFF
150	S3-3 ON, all others OFF
300	S3-4 ON, all others OFF
600	S3-5 ON, all others OFF
1200	S3-6 ON, all others OFF
2400	S3-7 ON, all others OFF
<del>4800</del> ***	S3-8 ON, all others OFF

\*Set no more than one switch to ON at the same time.

\*\*Rate required by standard 8-level TTY's (Teletype machine).

\*\*\*Assumes K-to-M jumper on Sol-PC is not installed. With K-M jumper in and L-M trace on back side of Sol-PC cut, SDI operates at ~~4800~~ 9600 Baud when S3-8 is ON and all others OFF.

NOTE FOR REV D Sol-PC BOARDS: With S3-7 ON and all others OFF, Baud rate is either 2400 (K-to-M jumper not installed) or 4800 (K-M jumper in and L-M trace on back side of Sol-PC cut). With S3-8 ON and all others OFF, Baud rate is 9600.

Table 7-3. Word Length Selection With S4-2 &amp; 3.

WORD LENGTH (Number of Bits)	SWITCH SETTINGS	
	S4-2	S4-3
5	ON	ON
6	ON	OFF
7	OFF	ON
8	OFF	OFF

## 7.5.11 Full/Half Duplex (F/H) Switch, S4-6

Set this switch to ON if you want half duplex operation in the terminal mode. In half duplex operation, data transmitted out the SDI (J1) is "looped back" and received by the SDI for subsequent

display on the monitor. Use this type of operation when your Sol works with an external computer that does not "echo" data back to the Sol.

For full duplex operation in the terminal mode, set S4-6 to OFF. Only received data is displayed in full duplex operation. Use full duplex when Sol's transmitted data need not be displayed. (Note that transmitted data from the Sol, if echoed back, is displayed as received data.)

#### NOTE

If no Baud rate is selected, data will not be transmitted out of the SDI.

#### 7.5.12 Keyboard

The keyboard is an output device that produces ASCII (American Standard Code for Information Interchange) encoded data. It is hard-wired to an input port on the Sol and is used for data entry. ASCII data is interpreted by the Sol as data and/or commands as determined by the current system monitor program. The monitor program may be in the personality module, ALS-8, Sol RAM memory or some memory.

### 7.6 THE KEYBOARD, GENERAL DESCRIPTION

The Sol Terminal Computer has ASCII 96-character keyboard. Its key arrangement conforms with the QWERTY (standard typewriter) format. As shown in the photo on page X-26, there are also 12 control keys (including five basic cursor controls) and seven special function keys. A 15-key arithmetic pad, available as an option on the Sol-10, is provided as standard equipment on the Sol-20.

#### 7.6.1 Operating Features

The Sol keyboard features N-key rollover. That is, several keys can be pressed at the same time without loss of characters or commands. Key entries, however, are in the order of actual key closures. (The keyboard circuitry includes a scanning circuit that prevents simultaneous key operation.)

#### 7.6.2 Keyboard Indicators


Three keys (SHIFT LOCK, UPPER CASE and LOCAL) have indicator lights to indicate keyboard/terminal status. When any of these keys is pressed to turn an indicator light on, the light remains on after the key is released to show that the status persists. Pressing the key again turns the light out to indicate the change in status.

## 7.7 INDIVIDUAL KEY DESCRIPTIONS

The exact function of most keys on the Sol keyboard is determined by the software used (e.g., the personality module). Others have predefined functions that are common to the CONSOL and SOLOS Personality Modules. (Note that any key that generates a code can be redefined by a program to perform a specific function.) The code generated by each key on the keyboard and the corresponding character, or symbol, produced by the Sol's character generator (U25) are given in Table 7-4 on Pages VII-18 through VII-21.

Table 7-4 has two main headings: 1) KEY which identifies the keys on the Sol keyboard and 2) HEXADECIMAL CODE/CHARACTER GENERATION which specifies for each key the hexadecimal code generated by the keyboard and the symbol produced by the Sol's character generator. The second heading is divided into three major categories: UNSHIFTED, SHIFTED and CONTROL. UNSHIFTED defines the results when operating the keys unshifted (lower case), SHIFTED provides the same information when they are operated shifted (upper case), and CONTROL defines the results of control sequences (refer to Paragraph 7.7.7 on Page VII-22). Within each of these three categories you will find the hexadecimal code generated and the symbol displayed in response to that code by either of the two possible character generators that can be supplied with your Sol, the 6574 and 6575. Some keys move the cursor without displaying a new character.

Looking at the "W" entry on Page VII-18 and reading across the table, we see that:

1. Pressing "W" unshifted would generate the code 77 and either character generator (6574 or 6575) produces a lower case "W" (w). Do not actually press the keys at this point.
2. Pressing "W" shifted would generate the code 57 and either character generator would produce an upper case "W" (W).
3. Pressing CTRL (control) and "W", whether shifted or unshifted, generates the code 17 which causes the 6574 to produce the graphic symbol  for the ASCII "end of transmission block" control character and the 6575 to produce a two-character mnemonic (Eb) for that same control character.

In the following paragraphs, each key function is described in terms of its role in the terminal mode only and assumes the control character display option is enabled and the LOCAL indicator light is on. Many key functions differ from these descriptions in SOLOS command modes BASIC/5, ALS-8, etc. As an aid to learning each key location, we suggest that you keep the keyboard photo, X-26, in view as you study these functions.

### 7.7.1 Alphanumeric-Punctuation-Symbol Keys

These keys enter the applicable character into the Sol.

## PROCESSOR TECHNOLOGY CORPORATION

## Sol OPERATING PROCEDURES

## SECTION VII

Table 7-4. Sol Keyboard Assignments.

KEY #		HEXADECIMAL CODE/CHARACTER GENERATION								
		UNSHIFTED			SHIFTED			CONTROL		
		Hex. Code	Symbol Displayed*		Hex. Code	Symbol Displayed*		Hex. Code	Symbol Displayed*	
			6574	6575		6574	6575		6574	6575
STANDARD KEYS										
ESCAPE	1B	None	None	1B	None	E <sub>C</sub>	1B	None	None	
1   !	31	1	1	21	!	!	01	┐	S <sub>H</sub>	
2   "	32	2	2	22	"	"	02	└	S <sub>X</sub>	
3   #	33	3	3	23	#	#	03	┌	E <sub>X</sub>	
4   \$	34	4	4	24	\$	\$	04	↖	E <sub>T</sub>	
5   %	35	5	5	25	%	%	05	⊗	E <sub>Q</sub>	
6   &	36	6	6	26	&	&	06	└	A <sub>K</sub>	
7   '	37	7	7	27	'	'	07	↶	B <sub>L</sub>	
8   (	38	8	8	28	(	(	08	↶	B <sub>S</sub>	
9   )	39	9	9	29	)	)	09	↓	H <sub>T</sub>	
∅	30	∅	∅	20	None	None	00	None	None	
-   =	2D	-	-	3D	=	=	0D	Return	Return	
^   ~	5E	^	^	7E	~	~	1E	▣	R <sub>S</sub>	
[   {	5B	[	[	7B	{	{	1B	None	None	
\   -	5C	\	\	7C	-	-	1C	▣	F <sub>S</sub>	
]   }	5D	]	]	7D	}	}	1D	▣	G <sub>S</sub>	
BREAK	None	None	None	None	None	None	None	None	None	
TAB	09	→	H <sub>T</sub>	09	→	H <sub>T</sub>	09	↑	H <sub>T</sub>	
Q	71	q	q	51	Q	Q	11	⊗	D <sub>1</sub>	
W	77	w	w	57	W	W	17	└	E <sub>B</sub>	
E	65	e	e	45	E	E	15	⊗	E <sub>Q</sub>	
R	72	r	r	52	R	R	12	⊗	D <sub>2</sub>	
T	74	t	t	54	T	T	14	⊗	D <sub>4</sub>	
Y	79	y	y	59	Y	Y	19	⊗	E <sub>M</sub>	
U	75	u	u	55	U	U	15	↶	N <sub>K</sub>	
I	69	i	i	49	I	I	09	↑	H <sub>T</sub>	

## PROCESSOR TECHNOLOGY CORPORATION

## Sol OPERATING PROCEDURES

## SECTION VII

Table 7-4. Sol Keyboard Assignments. (Continued)

KEY #	HEXADECIMAL CODE/CHARACTER GENERATION								
	UNSHIFTED			SHIFTED			CONTROL		
	Hex. Code	Symbol Displayed*		Hex. Code	Symbol Displayed*		Hex. Code	Symbol Displayed*	
		6574	6575		6574	6575		6574	6575
STANDARD KEYS (Continued)									
O	6F	o	o	4F	O	O	0F	⊙	S <sub>I</sub>
P	70	p	p	50	P	P	10	⊠	D <sub>L</sub>
@   \	40	@	@	60	\	\	00	None	None
RETURN	0D	←	C <sub>R</sub>	0D	←	C <sub>R</sub>	0D	Return	Return
LINE FEED	0A	Line Feed	Line Feed	0A	Line Feed	Line Feed	0A	Line Feed	Line Feed
CTRL	None	None	None	None	None	None	None	None	None
SHIFT LOCK	None	None	None	None	None	None	None	None	None
A	61	a	a	41	A	A	01	└	S <sub>H</sub>
S	73	s	s	53	S	S	13	⊙	D <sub>3</sub>
D	64	d	d	44	D	D	04	↘	E <sub>T</sub>
F	66	f	f	46	F	F	06	↙	A <sub>K</sub>
G	67	g	g	47	G	G	07	⊠	B <sub>L</sub>
H	68	h	h	48	H	H	08	↶	B <sub>S</sub>
J	6A	j	j	4A	J	J	0A	Line Feed	Line Feed
K	6B	k	k	4B	K	K	0B	↓	V <sub>T</sub>
L	6C	l	l	4C	L	L	0C	↘	F <sub>F</sub>
;   +	3B	;	;	2B	+	+	0B	↙	V <sub>T</sub>
:   *	3A	:	:	2A	*	*	0A	Line Feed	Line Feed
DEL   _	7F	None	None	5F	Delete	Delete	1F	⊠	U <sub>S</sub>
REPEAT	None	None	None	None	None	None	None	None	None
CTRL	None	None	None	None	None	None	None	None	None
UPPER CASE	None	None	None	None	None	None	None	None	None
SHIFT	None	None	None	None	None	None	None	None	None
Z	7A	z	z	5A	Z	Z	1A	\$	S <sub>B</sub>
X	78	x	x	58	X	X	18	X	C <sub>N</sub>
C	63	c	c	43	C	C	03	└	E <sub>X</sub>

\*See notes at end of this table, Page VII-21.

## PROCESSOR TECHNOLOGY CORPORATION

## Sol OPERATING PROCEDURES

## SECTION VII

Table 7-4. Sol Keyboard Assignments. (Continued)







KEY #	HEXADECIMAL CODE/CHARACTER GENERATION								
	UNSHIFTED			SHIFTED			CONTROL		
	Hex. Code	Symbol Displayed*		Hex. Code	Symbol Displayed*		Hex. Code	Symbol Displayed*	
		6574	6575		6574	6575		6574	6575
STANDARD KEYS (Continued)									
V	76	v	v	56	V	V	16		S <sub>Y</sub>
B	62	b	b	42	B	B	02		S <sub>X</sub>
N	6E	n	n	4E	N	N	0E		S <sub>O</sub>
M	60	m	m	40	M	M	0D	Return	Return
, <	2C	,	,	3C	<	<	0C		F <sub>F</sub>
. >	2E	.	.	3E	>	>	0E		S <sub>O</sub>
/ ?	2F	/	/	3F	?	?	0F		S <sub>I</sub>
SHIFT	None	None	None	None	None	None	None	None	None
LOCAL	None	None	None	None	None	None	None	None	None
Space Bar	20	None	None	20	None	None	20	None	None
ARITHMETIC PAD KEYS									
-	2D	-	-	2D	-	-	2D	-	-
*	2A	*	*	2A	*	*	2A	*	*
÷	2F	/	/	2F	/	/	2F	/	/
7	37	7	7	37	7	7	37	7	7
8	38	8	8	38	8	8	38	8	8
9	39	9	9	39	9	9	39	9	9
4	34	4	4	34	4	4	34	4	4
5	35	5	5	35	5	5	35	5	5
6	36	6	6	36	6	6	36	6	6
1	31	1	1	31	1	1	31	1	1
2	32	2	2	32	2	2	32	2	2
3	33	3	3	33	3	3	33	3	3
∅	30	∅	∅	30	∅	∅	30	∅	∅
.	2E	.	.	2E	.	.	2E	.	.
+	2B	+	+	2B	+	+	2B	+	+



Table 7-4. Sol Keyboard Assignments. (Continued)

KEY#	HEXADECIMAL CODE/CHARACTER GENERATION								
	UNSHIFTED			SHIFTED			CONTROL		
	Hex. Code	Symbol Displayed*		Hex. Code	Symbol Displayed*		Hex. Code	Symbol Displayed*	
		6574	6575		6574	6575		6574	6575
SPECIAL KEYS									
LOAD	8C	None	FF	8C	None	FF	8C	None	FF
MODE SELECT	80	None	None	80	None	None	80	None	None
↑	97	None	None	97	None	None	97	None	None
←	81	None	None	81	None	None	81	None	None
→	93	None	None	93	None	None	93	None	None
↓	9A	None	None	9A	None	None	9A	None	None
HOME CURSOR	8E	None	None	8E	None	None	8E	None	None
CLEAR	8B	None	None	8B	None	None	8B	None	None

#Vertical line between characters indicates dual character key.  
\*Character generated is displayable and transmittable. "None" means no code is generated or no symbol is displayed. Return is defined in Section 7.7.11, and line feed in Section 7.7.12, on page VII-24.

### 7.7.2 Space Bar

Pressing the Space Bar, shifted or unshifted, generates the ASCII space code (20) and moves the cursor one space to the right.

### 7.7.3 Arithmetic Pad Keys

Except for the division symbol key ( $\div$ ), these keys enter the applicable character into the Sol. The division symbol key enters a forward slash (/) character. SHIFT does not affect these keys.

The arithmetic pad is useful for entering large amounts of numerical data. Each key in the pad duplicates its corresponding numeric, period (decimal point), dash (minus), plus (addition), asterisk (multiplication) and forward slash (division) key in the "typewriter" group of keys. That is, pressing one of the pad keys does the same thing as pressing its corresponding key in the "typewriter" group.

## 7.7.4 ESCAPE Key

Pressing ESCAPE, shifted or unshifted, generates the ASCII escape character (1B). The character is displayed.

## 7.7.5 BREAK Key

Pressing BREAK, shifted or unshifted, forces the SDI output line to a space level for as long as the key is depressed. No character is displayed. (Some communications systems use this feature.)

## 7.7.6 TAB Key

Pressing TAB, shifted or unshifted, generates the ASCII horizontal tab character (09). The character is displayed.

## 7.7.7 Control (CTRL) Key

CTRL, shifted or unshifted, is used with alphanumeric, punctuation and symbol keys to initiate functions or generate the characters defined in Table 7-4. Table 7-5 defines the ASCII control characters. The characters in Table 7-5 are not always displayed on the video monitor.

A control sequence (e.g., CTRL plus J, which produces ASCII line feed) requires that CTRL be pressed first and held down while the other key or keys are pressed in sequence.

## 7.7.8 SHIFT Key and SHIFT LOCK Key/Indicator

The SHIFT key generates no code and is thus not displayed. It is interpreted as a direct internal operation, and when pressed specifically shifts the keyboard from lower case to upper case and from the lower to upper character on dual character keys as on a typewriter. The keyboard remains in upper case as long as SHIFT is held down.

Pressing SHIFT LOCK to turn the indicator light on electronically locks the SHIFT key in the upper case position. Again, no code is generated and no character is displayed. Pressing SHIFT returns the keyboard to lower case and causes the SHIFT LOCK indicator light to go out.

## 7.7.9 UPPER CASE Key/Indicator

Pressing this key, shifted or unshifted, to turn the indicator light on activates the upper case keyboard function so that all alphabetic characters entered from the keyboard, regardless of SHIFT key status, are transmitted as upper case characters. (Dual character keys, however, do respond to the SHIFT key.) With the indicator light on, the Sol keyboard essentially simulates a teletype (TTY) keyboard.

Pressing UPPER CASE to turn the indicator light off returns the keyboard to normal SHIFT key operation.

Table 7-5. Control Character Symbols and Definitions.

HEXADECIMAL CODE	SYMBOL GENERATED BY		DEFINITION
	6574 Generator	6575 Generator	
06	✓	AK	Acknowledge
07	⌂	BL	Bell
08	↶	BS	Backspace
18	⌂	CN	Cancel
0D	↑	CR	Carriage Return
11	⊙	D1	Device Control 1
12	⊙	D2	Device Control 2
13	⊙	D3	Device Control 3
14	⊙	D4	Device Control 4
7F	⊠	⊠	Delete
10	⊠	DL	Data Link Escape
17	T	EB	End of Transmission Block
1B	e	EC	Escape
19	●	EM	End of Medium
05	⊠	EQ	Enquiry
04	↯	ET	End of Transmission
03	┘	EX	End of Text
0C	↯	FF	Form Feed
1C	⊠	FS	File Separator
1D	⊠	GS	Group Separator
09	↯	HT	Horizontal Tab
0A		LF	Line Feed
15	✕	NK	Negative Acknowledge
00	⊠	NU	Null
1E	⊠	RS	Record Separator
1A	\$	SB	Substitute
01	┘	SH	Start of Heading
0F	⊙	SI	Shift In
0E	⊗	SO	Shift Out
02	T	SX	Start of Text
16	┘	SY	Synchronous Idle
1F	⊠	US	Unit Separator
0B	↓	VT	Vertical Tab

## 7.7.10 LOCAL Key/Indicator

The LOCAL key internally connects the SDI output to the SDI input and disables serial transmission. No character is displayed. Pressing LOCAL, shifted or unshifted, to turn the indicator light on sets Sol for local operation. Keyboard entries are not transmitted, but they are "looped back" to the SDI input for display. That is, Sol is not on "line". Pressing LOCAL to turn the light off ends local operation. This corresponds to the local/line operation of a TTY.

## 7.7.11 RETURN Key

Pressing RETURN, shifted or unshifted, generates the ASCII carriage return character (ØD), which is not displayed, and moves the cursor to the start of the line on which it resided prior to RETURN being depressed. (This is the same action as a TTY carriage return.) RETURN also erases all data in the line to the right of the cursor.

## 7.7.12 LINE FEED Key

Pressing LINE FEED, shifted or unshifted, generates the ASCII line feed character (ØA), which is not displayed, and moves the cursor vertically downward one line. (This is the same action as a TTY line feed.) Line feed action does not erase any data in the line to the right of the cursor.

## 7.7.13 LOAD Key

The LOAD key character is displayed, but the key is non-functional with CONSOL and SOLOS. The code generated by this key is 8C, and it may be used by a program to meet a specific need.

## 7.7.14 REPEAT Key

The REPEAT key generates no character and is consequently not displayed. Pressing REPEAT, shifted or unshifted, and another key at the same time causes the other key to repeat at an approximate rate of 15 times per second as long as both keys are held down. Pressing REPEAT at the same time as UPPER CASE performs a restart. See Section 7.5.2 on page VII-13.

## 7.7.15 MODE SELECT Key

Pressing this key, shifted or unshifted, generates the code 8Ø and causes Sol to enter the command mode.

## 7.7.16 CLEAR Key

Pressing CLEAR, shifted or unshifted, erases the entire screen and moves the cursor to its "home" position (upper left corner of the screen).

### 7.7.17 Cursor Control (HOME CURSOR and Arrows) Keys

Five keys control basic cursor movement. They are HOME CURSOR and the four keys with arrows. None are affected by SHIFT status, and none are displayed or transmitted.

Pressing HOME CURSOR moves the cursor to its home position--the first character position in the upper left corner of the screen.

To move the cursor up, down, left or right, press the applicable "arrow" key. Each time you press a key the cursor moves one unit in the direction you wish--one space horizontally or one line vertically. These keys may be used with REPEAT. The cursor will not move across any margin of the screen with these four keys.

## 7.8 BASIC OPERATIONS

### 7.8.1 Switching From Terminal To Command Mode

To switch from terminal to command mode, simply press the MODE SELECT key. Sol enters the command mode, issues a prompt character (>) and waits for a command input.

### 7.8.2 Switching From Command To Terminal Mode

To switch from command to terminal mode, press UPPER CASE, TERM and RETURN in that order. Sol enters the terminal mode and all keyboard data will be sent to the SDI output and all data received (including "looped back" data) will appear on the screen.

### 7.8.3 Entering Commands In The Command Mode

The various commands for CONSOL and SOLOS are described in Section IX of this manual and the SOLOS Users' Manual respectively.

You can place more than one command on the screen. For each command, use the arrowed cursor control keys to position the cursor at the start of a new line and begin the new command line with a prompt character (>).

A command is executed when you press the RETURN key, and all characters on the line to the left of the cursor are interpreted as the command. This means that if more than one command line is on the screen, you can execute any one of them as follows: position the cursor with the arrowed cursor control keys to the right of the desired command and press RETURN.

Should you make a mistake when entering a command, there are two ways to correct it:

(Paragraph 7.8.3 continued on Page VII-26.)

1. If you see the error immediately (the error is to the immediate left of the cursor), press the DEL key (unshifted) to erase the mistake. Then make the correction.
2. If the error is more than one character position to the left of the cursor, use the arrowed cursor control keys to position the cursor over the mistake. Then make the correction

#### 7.8.4 Keyboard Restart

To perform a keyboard restart, press the UPPER CASE and REPEAT keys at the same time. This key combination performs the same function as a power on initialization or setting the RST switch to ON. Use the keyboard restart to return to SOLOS/CONSOL from 1) a program which does not recognize the MODE SELECT key or 2) a program that is stuck in an endless loop.

### 7.9 Sol-PERIPHERAL INTERFACING

#### 7.9.1 Audio Cassette Recorders

Your Sol is capable of controlling one or two recorders. The interconnect requirements for one recorder were previously covered in Paragraph 7.4.1 in this section.

Since the Sol has only one audio input and one audio output jack, however, the interconnect requirements for two recorders are somewhat different than for one.

You will need two "Y" adapters, one to feed the single Sol audio output to the AUXILIARY input of two recorders and the other to feed the MONITOR output of two recorders to the single Sol audio input. (If you intend to use the Audio In and Out cables described in Paragraph 7.4.1 in this section, miniature phone jack-to-two miniature phone plug adapters are required.) Since the recorder outputs are most likely unbalanced, we also suggest that you incorporate 1000 ohm resistors in the MONITOR adapter as shown in Figure 7-5 on Page VII-29. Figure 7-5 also illustrates, in schematic form, how to connect two recorders to your Sol.

When using two recorders you may read or write to both under program control as well as read one tape while writing on the other.

If you intend to read one tape while writing on the other, however, you may have to disconnect the MONITOR plug from the write unit, with the need for disconnect being determined by the recorder design. The MONITOR disconnect must be made if the recorder has a



"monitor" output in the record mode. (Panasonic RQ-413S and RQ-309DS do, for example.)

NOTE 1

Recorders on which the "monitor" jack is labeled MONITOR usually provide a monitor output in the record mode. If the jack is labeled EAR or EARPHONE, the recorder usually does not provide a monitor output in the record mode.

NOTE 2

To determine if your recorder provides a monitor output in the record mode, install a blank tape, plug earphone into "monitor" jack and microphone into MICROPHONE jack, set recorder controls to record, and speak into microphone while listening with the earphone. If you hear yourself through the earphone, your recorder does provide a monitor output in the record mode.

Write Operations. Other than placing the recorder(s) in the record mode, loading the cassette(s) and making sure that the head(s) is on tape (not leader), no manual operations are needed to write on tape.

In the case of two recorders, however, Unit 1 and 2 must be specified in the SAVE command in order to select the desired recorder. A default selects Unit 1. Refer to your SOLOS Users' Manual for instructions on how to use tape commands.

Read Operations. In order to read a specific file on tape, you must start the tape at least two seconds ahead of that file. This delay allows the Sol audio cassette interface circuitry and the recorder playback electronics to stabilize after power is turned on. Since all file searches are in the forward direction, the simplest approach is to fully rewind the cassette(s) before a read operation unless you know that the file of interest is advanced at least two seconds. (See Paragraph 7.4.3, Step 21 for instructions on how to rewind the tape.)

For a read operation, proceed as follows:

1. Load cassette(s) as just described.
2. If only one recorder is used, set its volume control at midrange. With two recorders, set both volume controls at their high end.

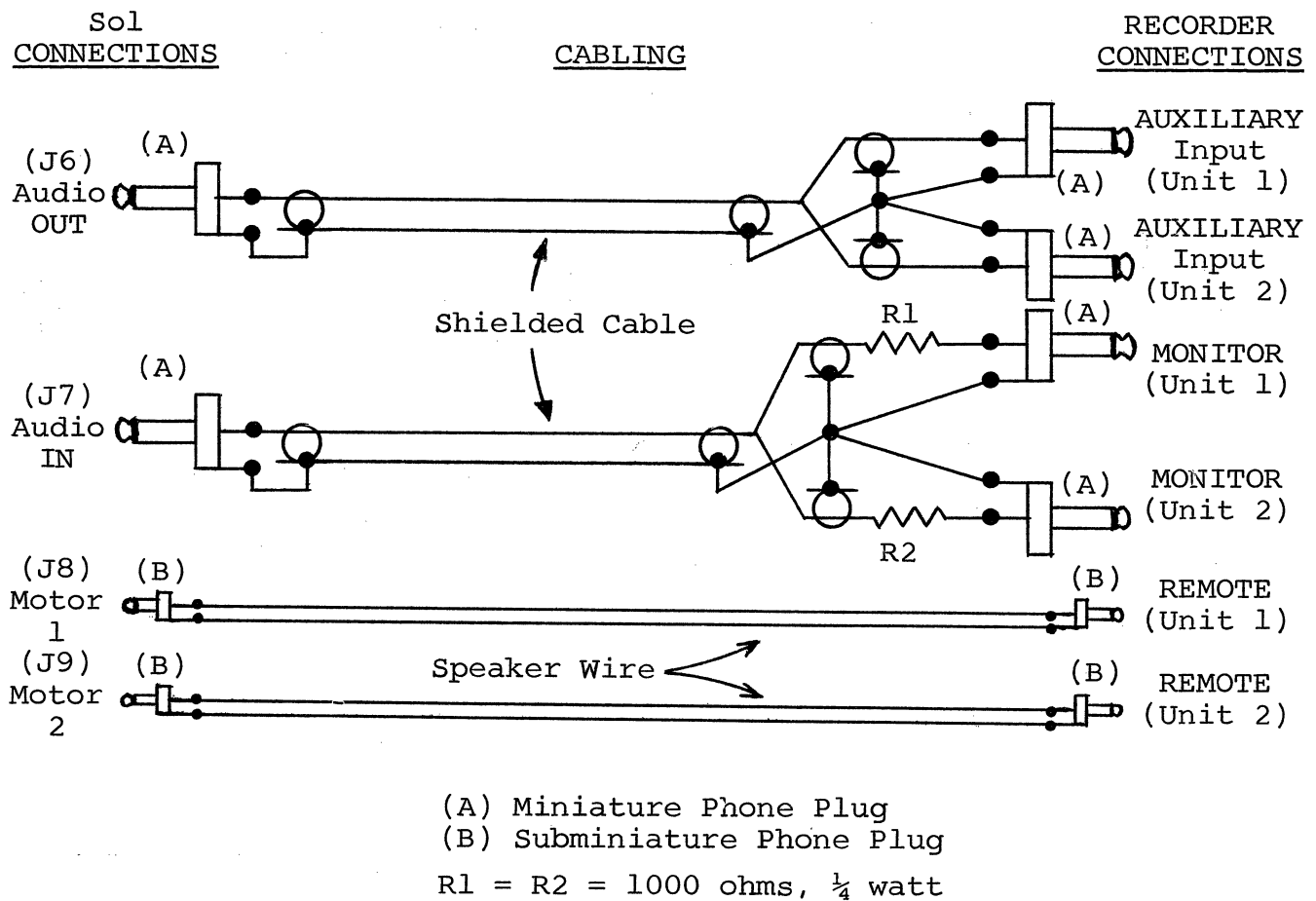


Figure 7-5. Connecting Sol to two cassette recorders.

3. Set recorder(s) tone control(s) at the top of the range (maximum treble).
4. Set PLAY control(s) for playback mode.
5. Give Sol the GET or "GET, then Execute" command as appropriate. (Refer to your SOLOS Users' Manual for instructions on how to use tape commands.)

#### 7.9.2 Serial Data Interface (SDI)

The Sol Serial Data Interface (J1) is capable of driving an RS-232 device, such as a modem, or a current loop device, such as the ASR33 TTY.

S3 (Baud Rate) and S4 (Parity, Word Length, Stop Bits and Full/Half Duplex) are used to select the various serial interface options as described in Paragraphs 7.5.7 through 7.5.11 in this section.

Set S3 switches to select the Baud rate required by the modem or current loop device. (Standard 8-level TTY's operate at 110 Baud, S3-2 ON and all other S3 switches OFF.) For standard 8-level TTY's and most modems, set all S4 switches OFF. (This selects eight data bits, two stop bits, no parity bit and full duplex operation for the SDI.

Figures 7-6 and 7-7 show examples of current loop and modem interconnections to the Sol SDI connector (J1). The ASR33 TTY is used to illustrate a current loop interconnect, and the Bell 103 modem is used to illustrate a modem interconnect.

When operating in the terminal mode and full duplex, Sol keyboard data is transmitted out on Pin 2 of J1 and data received on Pin 3 of J1 is displayed on the video monitor. In the command mode, SOLOS set in and out commands can be used to channel output data and input data through the SDI. (Refer to your SOLOS Users' Manual for instructions on how to use the set commands.)

In either mode, the LOCAL key directly controls the SDI. With the LOCAL indicator light on, received data is ignored and keyboard data is not transmitted. It is, however, looped back for display on the video monitor. With the LOCAL light off, received data is displayed and keyboard data is transmitted but not displayed unless it is echoed back.

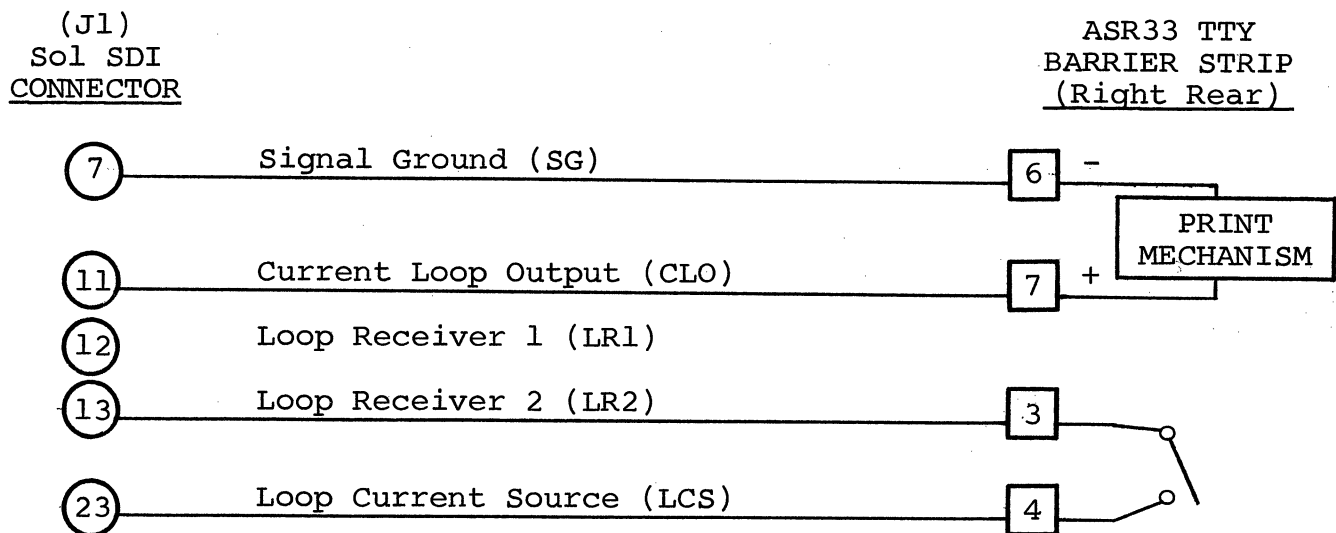
### 7.9.3 Parallel Data Interface (PDI)

The Sol Parallel Data Interface (J2) is used to drive parallel devices such as paper tape readers/punches and line printers. It provides eight output data lines, eight input data lines, four handshaking signals and three control signals. The latter allow up to four devices to share the PDI connector. (See Appendix VII for J2 pinouts.)

The port address for parallel input and output data is FD (hexadecimal), and the control port address for the PDI is FA (hexadecimal). PXDR is available at bit 2 of port FA. When this bit is set to 0, the external device is ready to receive a byte of data. PDR is available at bit 1 of port FA, with 0 indicating the external device is ready to send a byte of data. Parallel Unit Select (PUS) is controlled by bit 4 of port FA. The input and output enable lines are available for tri-stating an external two-way data bus.

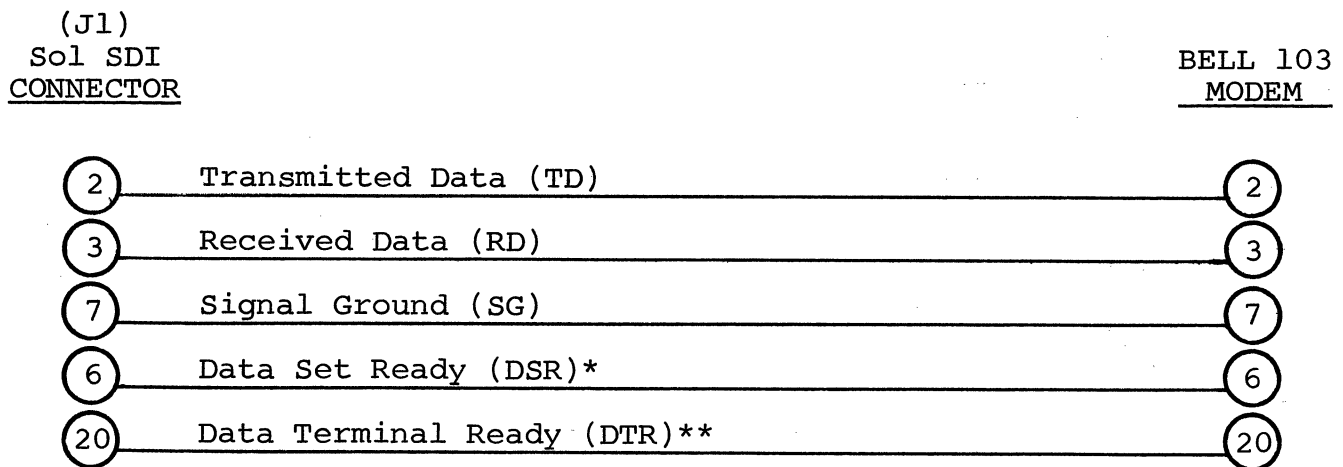
Use of the three control signals is optional and is unnecessary when only one device is connected to the PDI connector.

(Paragraph 7.9.3 continued on Page 31.)



CAUTION: PINS 1 AND 2 ON TTY BARRIER STRIP  
CARRY 120 V ac LINE VOLTAGE.

Figure 7-6. Connecting Sol SDI to current loop device such as TTY.



\*Available at bit 1 of port F8. Terminal mode software (SOLOS et al) does not use this signal and transmits data whether or not the modem is ready.

\*\*Sol is wired so that DTR indicates a ready condition whenever power is on.

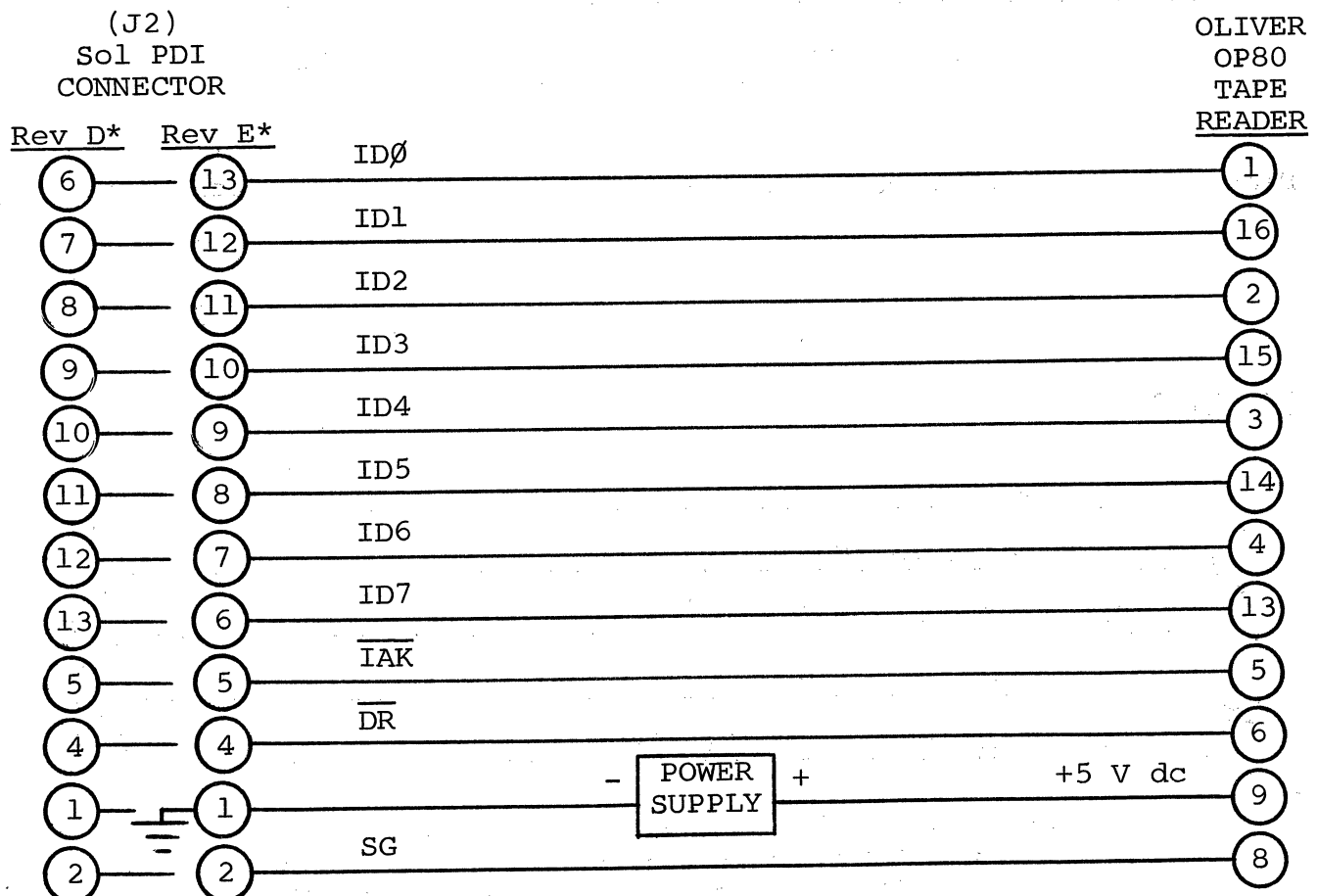
Figure 7-7. Connecting Sol SDI to communications modem.

In Figure 7-8, the Oliver OP80 Manual Paper Tape Reader is used to illustrate a typical PDI interconnect.

### 7.10 CHANGING THE FUSE

Sol is protected with a 3.0 amp Slo-Blo fuse housed on the rear panel (see Figure 7-1 on Page VII-6). To remove the fuse, turn Sol off, disconnect power cord, turn fuse post cap one quarter turn counterclockwise, pull straight out and remove fuse from cap.

To install a fuse, insert fuse in cap, push in and turn one-quarter turn clockwise.



NOTE: +5 V dc is not available at J2. The use of an external +5 V dc power supply with its ground connected to Pin 1 of J2 (Sol chassis ground) is recommended.

\*Sol-PC Board

Figure 7-8. Connecting Sol PDI to parallel device.



## VIII THEORY OF OPERATION

8.1	INTRODUCTION . . . . .	VIII-1
8.2	OVERVIEW . . . . .	VIII-1
8.3	BLOCK DIAGRAM ANALYSIS, Sol-PC . . . . .	VIII-3
8.3.1	Functional Elements And Their Relationships	VIII-3
8.3.2	Typical System Operation . . . . .	VIII-5
	Keyboard Data Entry and Display . . . . .	VIII-5
	SDI/UART Transfer and Display . . . . .	VIII-6
8.4	POWER SUPPLY CIRCUIT DESCRIPTION . . . . .	VIII-6
8.5	Sol-PC CIRCUIT DESCRIPTIONS . . . . .	VIII-8
8.5.1	CPU and Bus . . . . .	VIII-8
8.5.2	Memory and Decoder . . . . .	VIII-14
8.5.3	Input/Output . . . . .	VIII-18
8.5.4	Display Section . . . . .	VIII-22
8.5.5	Audio Tape I/O . . . . .	VIII-33
8.6	KEYBOARD . . . . .	VIII-38
8.6.1	Block Diagram Analysis . . . . .	VIII-38
8.6.2	Circuit Description . . . . .	VIII-39

## TABLES AND ILLUSTRATIONS

<u>TABLE</u>	<u>TITLE</u>	<u>PAGE</u>
8-1	Port Decoder (U35 & U36) Outputs and Their Functions . . . . .	VIII-17

<u>FIGURE</u>	<u>TITLE</u>	<u>PAGE</u>
8-1	Clock Generator Timing . . . . .	VIII-11
8-2	Example of uppercase character (I) display . . . . .	VIII-24
8-3	Example of lowercase character (p) display . . . . .	VIII-24
8-4	Video Display timing . . . . .	VIII-27
8-5	6574 Character Generator ROM pattern . . . . .	VIII-30
8-6	6575 Character Generator ROM pattern . . . . .	VIII-31



## 8.1 INTRODUCTION

This section concerns itself with the hardware aspects of the Sol Terminal Computer<sup>TM</sup>. It specifically deals with the operation of the power supply and the logic associated with the Sol-PC and keyboard. Descriptions of software and the operation of the circuitry contained in the multitude of integrated circuits (IC's) used in the Sol fall outside the scope of this section. In some cases, references to other publications or sections in this manual are provided when it is felt that additional information will contribute to a better understanding of how Sol operates. Should the reader wish to delve further into the operation of a specific IC, we suggest that he study the appropriate data sheet for that IC.

The section begins with an overview of the Sol design. A block diagram analysis then provides the reader with an understanding of the relationship between the functional elements of the Sol-PC. This analysis sets the stage for detailed descriptions of the circuitry that makes up these elements. The section concludes with a block diagram analysis and circuit description of the keyboard.

## 8.2 OVERVIEW

The Sol Terminal Computer<sup>TM</sup>, as the name implies, is both a terminal and computer. It is designed around the S-100 bus structure used in other 8080 microprocessor-based computers and incorporates all of the circuitry needed to perform either function. In essence, Sol combines a central processor unit (CPU) with several S-100 peripheral modules--memory, keyboard input interface (including the keyboard), video display output interface plus audio cassette tape, parallel, and serial input/output (I/O) interfaces. Sol-20 also includes a five-slot backplane board for adding other memory and I/O modules that are compatible with the S-100 bus.

An 8080 microprocessor (the CPU) is the "brain" of the Sol. It controls the functions performed by the other system components, obtains (fetches) instructions stored in memory (the program), accepts (inputs) data, manipulates (processes) data according to the instructions and communicates (outputs) the results to the outside world through an output port. (For information on 8080 operation, refer to the "Intel® 8080 Microcomputer Systems User's Manual.")

As shown in the Sol Simplified Block Diagram on Page X-24 in Section X, data and control signals travel between the CPU and the rest of the Sol over three buses: 1) a 16-line Address Bus, 2) an eight-line Bidirectional Data Bus, and 3) a 28-line Control Bus which is interfaced to the CPU with support logic circuitry. (Note that the use of a bidirectional data bus permits eight lines to do the work of 16, eight input and eight output.) These three buses account for the bulk of the S-100 Bus which connects the Sol to expansion memory and I/O modules.

In the Sol-20, the S-100 Bus structure takes the form of a five-slot backplane board. It consists of a printed circuit board with 100 lines (50 on each side) and five edge connectors on which like-numbered pins are connected from one connector to another. Functionally, the Sol version of the S-100 Bus is comprised of:

1. Sixteen output address lines from the CPU which are input to all external memory and I/O circuitry. (Direct memory access (DMA) devices must generate addresses on these lines for DMA transfers.)
2. Eight data input/output lines that transfer data between external memory and I/O devices and the CPU or DMA devices. (These eight lines are paralleled with eight other bus lines.)
3. Eight status output lines from the CPU support logic: Memory and I/O devices use status signals to obtain information concerning the nature of the CPU cycle. (DMA devices must generate these signals for DMA transfers.)
4. Nine processor command and control lines: Six of these are output signals from the CPU support logic; three of them are input signals to the CPU support logic from memory and I/O devices. (In a DMA transfer, the DMA device assumes control of these lines.)
5. Five disable lines: Four of these are supplied by a DMA device to disable the tri-state drivers on the CPU outputs during DMA transfers. The fifth is a derivative of the DBIN output from the CPU, and it is used to disable any memory addressed in Page 0. Use of this disable is optional with a jumper.
6. Two input lines to the CPU support logic which are used for requesting a wait period. One is used by memory and I/O devices and the other by external devices.
7. Six power supply lines which supply power to expansion modules.
8. Three clock lines.
9. Four special purpose signal lines.
10. Thirty-one unused lines.

Definitions for each S-100 Bus line, as used in the Sol, are provided on Pages AVII-3 through AVII-6 in Appendix VII.

In addition to the S-100 Bus structure, Sol also uses an eight-line keyboard input port, an eight-line parallel input port,

an eight-line parallel output port, an eight-line sense switch logic input port, and a unidirectional eight-line internal data bus.

The use of a unidirectional (input) data bus accommodates Sol's internal low-drive memory and I/O devices that do not meet the heavy drive requirement of the bidirectional data bus. The low-drive requirement of the internal bus also allows using the tri-state capabilities of the UART's (Universal Asynchronous Receiver/Transmitter) in the serial and audio cassette I/O circuits without additional drivers.

All CPU data and address lines are buffered through tri-state drivers to support a larger array of memory and I/O devices than would otherwise be possible with the 8080 output drive capability. Data input to the CPU is selected by a four-input multiplexer from the Keyboard Port, Parallel Port, Bidirectional Data Bus and Internal Data Bus. The Internal Data Bus is the source of all data input to the CPU from Sol's internal memory, the serial interface and the cassette interface. The Bidirectional Data Bus is the source of all data fed to memory and I/O, both internal and external. It is also the source of data input to the CPU from eight internal sense switches as well as from external memory and I/O.

### 8.3 BLOCK DIAGRAM ANALYSIS, Sol-PC

#### 8.3.1 Functional Elements And Their Relationships

As can be seen in the Sol block diagram on Page X-24 in Section X, timing signals for Sol are derived from a crystal controlled oscillator that produces a "dot clock" frequency of 14.31818 MHz. (This frequency, four times that of the NTSC color burst, provides compatibility with color graphics devices.) The dot clock is applied directly to the Video Display Generator circuit and divided in the Clock Generator to provide  $\phi 1$ ,  $\phi 2$  and CLOCK. CLOCK synchronizes all control inputs to the 8080;  $\phi 1$  and  $\phi 2$  are the nonoverlapping, two phase clocks required by the 8080.

Memory internal to the Sol is divided between 2K of ROM (Read Only Memory), 1K of System RAM (Random Access Read/Write Memory) and 1K of Display RAM. The ROM permanently stores the instructions that direct the CPU's activities. (To enhance Sol's versatility, this particular memory is on a plug-in "personality module". Thus, Sol can be easily optimized for a particular application by plugging in a personality module that contains a software control program designed for the task. The CONSOL and SOLOS programs, which are described in Section IX, are examples of such personality modules.) Display RAM stores data for display on a video monitor, and the System RAM provides temporary storage for programs and data. All memories are addressed on the Address Bus (ADR $\emptyset$ -15) and, except for the Display RAM, input data to the CPU on the Internal Data Bus (INT $\emptyset$ -7). Data entry into both RAM's is done on the Bidirectional Data Bus (DIO $\emptyset$ -7).

As can be seen, Sol's internal memory consists of four contiguous 1024-byte pages. There are two pages (C0 and C4, hexadecimal or hex) of ROM, with Page C0 at hex addresses C000 through C3FF and Page C4 at hex addresses C400 through C7FF. System RAM (Page C8) is at hex addresses C800 through CBFF, and Display RAM (Page CC) is at hex addresses CC00 through CFFF.

The six high order bits of the address are decoded in the Address Page and I/O Port Decoder to supply the required four memory page selection signals. The I/O Port Decoder portion of this circuit decodes the eight high order address bits to provide outputs that control Data Input Multiplexer switching, Data Bus Driver enablement and I/O port selection.

The video display section consists of the Video Display Generator and Display RAM. The RAM is a two-port memory, with the CPU having the higher priority. Screen refresh circuitry in the Video Display Generator controls the second port to call up data as needed for conversion by a character generator ROM into video output signals. Other circuitry generates horizontal and vertical sync and blanking signals as well as cursor and video polarity options.

A 1200 Hz signal, extracted from dot clock by a divider in the Video Display Generator, drives the Baud Rate Generator. This generator supplies the receive and transmit clocks for the serial data interface (SDI/UART) and provides all frequencies required for Baud rates between 75 and 9600. It also supplies clock signals to the Cassette Data Interface (CDI).

A UART controls data flow through the Serial Data Interface (SDI/UART) and provides for compatibility between the Sol and a data communications system, be it RS-232 standard or a 20 ma current loop device. In the transmit mode, parallel data on the Bidirectional Data Bus is converted into serial form for transmission. Received serial data is converted in the receive mode into parallel form for entry into the CPU on the Internal Data Bus. SDI/UART status is also reported to the CPU on the Internal Data Bus. The SDI/UART channel is enabled by the port strobe from the Address Page and I/O Port Decoder.

Circuitry within the CDI derives timing signals from clocks supplied by the Baud Rate Generator. The Cassette Data UART functions to 1) convert parallel data on the Bidirectional Data Bus into serial audio signals for recording on cassette tape, and 2) convert serial audio signals from a cassette recorder into parallel data for entry into the CPU from the Internal Data Bus. Note that Cassette Data UART status is also reported to the CPU on the Internal Data Bus. Again, a UART performs the necessary parallel-to-serial and serial-to-parallel conversions. Other CDI circuitry performs the needed digital-to-audio and audio-to-digital conversions and provides the signals that allow motor control for two recorders. As with the SDI/UART, the Cassette Data UART is enabled by a port strobe from the Address Page and I/O Port Decoder.

Output data from the CPU that is channeled through the Parallel Port (PP) is latched from the Bidirectional Data Bus by the parallel strobe from the Address Page and I/O Port Decoder. This data is made available at P2, the PP connector. Parallel input data (PID0-7) on P2, however, is fed directly to the Data Input Multiplexer for entry into the CPU.

As can be seen, keyboard data (KBD0-7) from J3 is also fed directly to the Data Input Multiplexer. The keyboard data ready flag, though, is input to the CPU on the internal data bus.

The remaining internal source of data input to the CPU is the Sense Switch Logic, with the data being input on the Bidirectional Data Bus. This is an eight-switch Dual Inline Package (DIP) array that lets the CPU read an eight-bit word when it issues the sense switch strobe via the Address Page and I/O Port Decoder. The sense switch data source is available to interact with the user's software.

CPU Support Logic accepts six control outputs from the CPU, status information from the CPU's data bus and control signals from the Control Bus. It controls traffic on the data buses by generating signals to 1) select the type of internal or external device (memory or I/O) that will have bus access and 2) assure that the device properly transfers data with the CPU.

### 8.3.2 Typical System Operation

Basic Sol system operation is as follows: The CPU fetches an instruction and in accordance with that instruction issues an activity command on the Control Bus, outputs a binary code on the Address Bus to identify the memory location or I/O device that is to be involved in the activity, sends or receives data on the data bus with the selected memory location or I/O device, and upon completion of the activity issues the next activity command.

Let's now look at some typical operating sequences.

Keyboard Data Entry and Display. Assume the "A" and SHIFT keys on the keyboard are pressed. The keyboard circuitry converts the key closures into the 7-bit ASCII (American Standard Code for Information Interchange) code for an "A" (1000001) and sends a keyboard-data-ready status signal to the CPU on the Internal Data Bus. The monitor program in ROM repetitively "looks" for the status signal. When it finds this signal the program enters its keyboard routine and enables the transfer by switching the Data Input Multiplexer to the keyboard bus via the Address Page and I/O Port Decoder.

Following program instructions, the CPU addresses the Display RAM on the Address Bus to determine where the next character is to appear on the screen. It then stores the ASCII code for the "A" at the appropriate location in the Display RAM and adds one to the cursor position in readiness for the next character. (Addressing is

done over the Address Bus; cursor position and the "A" enter the Display RAM on the Bidirectional Data Bus.) The CPU is now finished with the transfer, and will issue the next activity command.

When the refresh control circuitry calls up (addresses) the "A" from the Display RAM, the character generator ROM decodes the ASCII-coded "A" that is input from the Display RAM and generates the "A" dot pattern (see Figure 8-5 and 6) in parallel form. The ROM output is serialized into a video signal and combined with a composite sync signal to provide an Electronic Industries Association (EIA) composite video signal for display on an external video monitor.

SDI/UART Transfer and Display. A data transfer through the SDI/UART is similar to a keyboard entry, but data can be transferred in either direction.

Assume the SDI/UART wants to transfer an "A" from a modem to the CPU for display on a video monitor. The ASCII code for the "A", received in serial form from the modem on the serial data input of the SDI connector (J1), is fed to the SDI/UART. In the receiver section of the UART the serial data is converted into parallel form and placed in the UART's output register. The UART also sends a "received data ready" status signal to the CPU on the Internal Data Bus. When the program in ROM checks and finds the status signal, the program enters the SDI routine, and enables the transfer by switching the Data Input Multiplexer to the Internal Data Bus. The "A" enters the CPU on the Internal Data Bus and is sent to the Display RAM on the Bidirectional Data Bus. Operations involved in displaying the "A" are identical to a keyboard entry.

Now assume the CPU wants to send an "A" to the SDI/UART for transmission. The CPU, under program control, sends the SDI/UART status input port strobe via the Address Page and I/O Port Decoder to the UART. In turn, the UART responds with its status on the Internal Data Bus. Assuming the UART is ready to transmit, the CPU places the ASCII code for the "A" on the Bidirectional Data Bus and sends the SDI/UART data output port strobe which loads the Bidirectional Data Bus content into the UART's transmitter section. The "A" is serialized by the UART and sent out the transmitted data pin of J1.

#### 8.4 POWER SUPPLY CIRCUIT DESCRIPTION

Refer to the Sol-REG and Sol-10 or Sol-20 Power Supply Schematics in Section X, Pages X-12, 13 and 14.

The Sol power supply consists of the Sol-REG regulator and either the Sol-10 or Sol-20 power supply components. An 8 V dc unregulated supply in the Sol-20 is the only difference between the two. We will, therefore, describe the complete Sol-10 supply followed by the unregulated 8 V dc supply in the Sol-20.



Fused primary power is applied through S5 to T1 (T2 in the Sol-20). FWB1, a full-wave bridge rectifier, is connected across the 8-volt secondary (green leads). The rectified output is filtered by C8 and applied to the collector of Q1. Q1, a pass transistor, is driven by Q2, with the two connected as a Darlington pair. The output of Q1 is connected to R1 which serves as an overload current sensor.

An overload current (approximately 4 amps) increases the voltage drop across R1. The difference is amplified in one-half of U2 (an operational amplifier) and the output on pin 7 turns Q3 on. Q3 in turn "steals" current from Q1-Q2 and diverts current from the output on pin 1 of U2. This in effect turns the supply off to reduce the current and voltage. Note that the circuit is not a constant current regulator since the current is "folded back" by R6 and R8. The current is reduced to about 1 amp as the output voltage falls to zero.

Divider network R11 and R12, which is returned to -12 volts, senses changes in the output voltage. If the output voltage is 5 volts, the input on pin 2 of U2 is at zero volts. U2 provides a positive output on pin 1 if pin 3 is more positive than pin 2 and a negative output for the opposite condition.

When the output voltage falls below 5 volts, pin 2 of U2 goes more negative than pin 3. This means pin 1 of U2 goes positive to supply more current to the base of Q1. The resulting increase in current to the load causes the output voltage to rise until it stabilizes at 5 volts. Should the output voltage rise above 5 volts, the circuit operates in a reverse manner to lower the voltage.

Protection against a serious over-voltage condition (more than 6 volts) is provided by SCR1, D1, R2, R13, R14 and C8. Zener diode, (D1), with a 5.1 zener voltage, is connected in series with R13 and R2. When the output voltage exceeds about 6 volts, the resulting voltage drop across R2 triggers SCR1 to short the foldback current to ground. Since the overload current circuit is also working, the current through SCR1 is about 1 amp. Once the current is removed, this circuit restores itself to its normal condition; that is, SCR1 turns off. R13, R14 and C8 serve to slightly desensitize the circuit so that it will not respond to small transient voltage spikes.

Bridge rectifier FWB2, connected across the other T1 secondary, supplies +12 and -12 V dc. The positive output of FWB2 is filtered by C5 and regulated by IC regulator U1. The negative output is filtered by C4 and regulated by U3. Shunt diodes D3 and D4 protect U1 and U3 against discharge of C6 and C7 when power is turned off. (Note that should the -12 volt supply short to ground, the +5 volt supply turns off by the action of U2.)

Unregulated -16 and +16 V dc, at 1 amp, from the filtered outputs of FWB2 are made available on terminals X6 and X5. These are not used in the Sol-10, but they are supplied to the backplane board in the Sol-20 to drive S-100 Bus modules.

In the case of the Sol-20, the power transformer (T2) has an additional 8-volt secondary winding and a third bridge rectifier (FWB3) to supply +8 V dc at 8 amps. The output of FWB3 is filtered by C9 and controlled by bleeder resistor R13. Again, this voltage is supplied to the backplane board in the Sol-20.

Sol-20 also includes a cooling fan powered by the AC line voltage.

## 8.5 Sol-PC CIRCUIT DESCRIPTIONS

### 8.5.1 CPU and Bus

Refer to the CPU and Bus Schematic in Section X, Page X-15.

A crystal, two inverter sections in U92 and four D flip-flops (U90) and associated logic make up the Clock Generator.

The two U92 sections function as a free-running oscillator that runs at the crystal frequency of 14.31818 MHz. R133 and R134 drive these two sections of U92 into their linear regions, and C61 and 64 provide the required feedback loop through the crystal. U77, a permanently enabled tri-state non-inverting buffer/amplifier, furnishes a high drive capability.

This fundamental clock frequency is fed directly to the Video Display Generator and to the clock inputs of U90. U90 is a four-stage register connected as a ring counter that is reset to zero when power is applied to the Sol. This reset is accomplished with D8, R104 and C39.

The bits contained in the ring counter shift one to the right with every positive-going clock transition, but the output of the last stage is inverted or "flipped" before being fed back to the input. In a simple four-stage "flip-tail" ring counter, the contents would progress from left to right as follows: 1000, 1100, 1110, 1111, 0111, 0011, 0001, 0000--on the first through eighth clocks respectively. The hypothetical counter would go through eight states, dividing the clock by eight.

The Sol counter, however, is a modified flip-tail ring counter that can be configured to divide by one of three divisors--5, 6 or 7. This is made possible by using a two-input NAND gate (U91) in the feedback path and three jumper options (no jumper, D-to-C and D-to-E) to alter the feedback path. Let's see how it works.

Sol is normally configured with the D-to-E jumper installed to meet the clock requirements of the 8080A CPU. With this jumper installed, the outputs of the third and fourth U90 stages are applied to pins 9 and 10 of U91. Assuming U90 is reset to zero, pin 8 of U91 is high, and on the first clock pulse the counter contents change to 1000. (Refer to 2.045 MHz Clocks portion of Figure 8-1 on Page VIII-11.) Pin 8 of U91 cannot change until the fourth state (1111), at which time it goes to zero. On the fifth clock pulse the counter changes to 0111. Again, pin 8 of U91 cannot change from zero until one of its inputs changes. As shown in Figure 8-1, the third U90 stage (C) changes on the seventh clock. The counter now stands at 0001, and on the eighth clock the counter flips to 1000 and the count cycle repeats. The pattern is thus 1000, 1100, 1110, 1111, 0111, 0011, 0001. U90 consequently goes through seven states. We have a 3.5-stage counter that divides DOT CLOCK by seven to supply a 2.045 MHz output.

With no jumper installed, pin 10 of U91 is pulled high by R105, and U91 operates as a simple inverter for feeding back the output of the third U90 stage. In effect we have a three-stage counter that operates in a similar manner to that described in the preceding paragraph. It goes through six states (100, 110, 111, 011, 001, 000) to divide DOT CLOCK by six which produces a 2.386 MHz output. The timing for this option is also shown in Figure 8-1.

Let's now put the D-to-C jumper in. The feedback in this case is the NAND combination of the outputs from the second (B) and third (C) U90 stages. This gives us a 2.5-stage counter that divides DOT CLOCK by five. As can be determined from the 2.863 MHz portion of Figure 8-1, the counter has five states with this option, and the count pattern is: 100, 110, 111, 011, 001.

Outputs from U90 are applied to the logic comprised of the remaining three sections in U91. This logic and the A-to-B jumper option permits extracting clock pulses of varying widths and relationships to each other from various points within the counter. We extract two clock signals: 01 on pin 6 of U91 and 02 on pin 11 of U91. (The ability to select the frequency and pulse width for 01 and 02 permits the use of either the 8080A, 8080A-1 or 8080A-2 CPU for U105. The "A" version is the slowest speed unit, the "A-2" has an intermediate speed, and the "A-1" is the fastest.) Let's now see how the pulse width of 01 and 02 are determined.

01 on pin 6 of NAND gate U91 is low only when its two inputs are high, and this happens only when there is a 1 in the second and fourth stages of U90. This occurs during the time between the fourth and sixth fundamental clocks for 2.04 MHz operation--the fourth and fifth clocks for 2.38 MHz and 2.86 MHz. Keeping in mind that the fundamental clock period is 70 nsec, it is readily seen that the low frequency pulse train on pin 6 of U91 has a pulse width of 140 nsec and the two higher frequency pulse trains have a pulse width of 70 nsec. (Refer to Figure 8-1 on Page VIII-11.)

The A-to-B jumper is installed when the 8080A or 8080A-1 CPU is used in the Sol. Note that the output ( $\phi 2$ ) on pin 11 of NAND gate U91 is low only when the output on pin 3 of NOR gate U91 is high. (This section in U91 is actually a two-input NAND gate which is functionally the same as a two-input NOR gate.) Pin 3 of U91, with the A-to-B jumper in, is high when either the second (B) or third (C) U90 stage is at zero. As shown in Figure 8-1, this occurs between the sixth and tenth DOT CLOCKS, or 280 nsec ( $4 \times 70$  nsec), for 2.04 MHz operation. For 2.863 MHz, it occurs between the fifth and eighth DOT CLOCKS for 210 nsec. The section of NAND gate U91 with its output on pin 11 inverts the output on pin 3 of U91 and introduces a slight delay to insure there is no overlap between  $\phi 1$  and  $\phi 2$ .

With the A-to-B jumper out, pin 11 of U91 is low only when the second stage (B) of U90 is at zero. At 2.386 MHz, this occurs between the fifth and eighth DOT CLOCKS for 210 nsec. This configuration is used for the 8080A-2 CPU.

In summary, we have two non-overlapping pulse trains which represent the  $\phi 1$  and  $\phi 2$  clocks required by the 8080 CPU, and the pulse widths of these two clocks vary with frequency as follows:

<u>FREQUENCY</u>	<u><math>\phi 1</math> PULSE WIDTH</u>	<u><math>\phi 2</math> PULSE WIDTH</u>	<u>CPU</u>
2.045 MHz	140 nsec	280 nsec	8080A
2.386 MHz	70 nsec	210 nsec	8080A-2
2.863 MHz	70 nsec	210 nsec	8080A-1

$\phi 1$  and  $\phi 2$  are applied to S-100 Bus pins 25 and 24 respectively through inverters (U92) and bus drivers (U77). They are also capacitively coupled to pins 2 and 4 respectively of driver U104, the phase clock conditioner.

An additional clock, called CLOCK, is taken from pin 8 of NAND gate U91. It occurs 70 nsec after  $\phi 2$ . It is used on the Sol-PC and is also made available on S-100 Bus pin 49 as a general 2.04, 2.38 or 2.86 MHz clock signal.

Three J-K flip-flops (U63 and 64) are used to synchronize the READY, RESET and HOLD inputs to the CPU. All three are connected as D-type flip-flops so that their outputs follow their inputs on the low-to-high transition of the clock. The READY flip-flop input on pins 2 and 3 of one section in U63 is either PRDY or XRDY from the S-100 Bus; these are normally pulled high by R34 and R12 respectively. S-100 Bus signal PRESET, which is normally pulled high by R55, inputs the RESET flip-flop, the other section of U63. The HOLD flip-flop (U64) input is  $\overline{P}$  HOLD, normally pulled high by R56, from the S-100 Bus. Pull up resistors R51, R50 and R53 insure that the high states of these three flip-flops are adequate for the CPU.

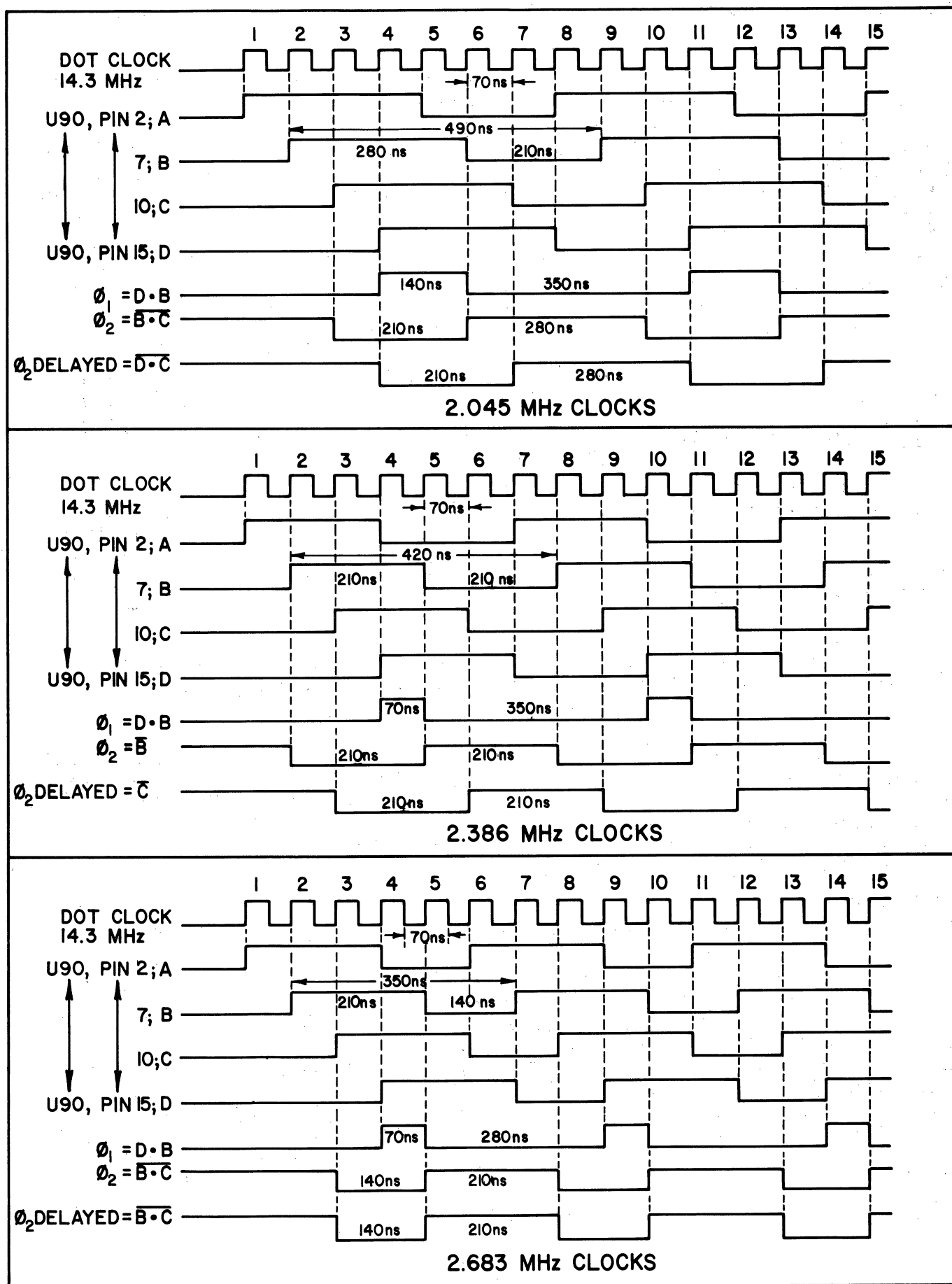


Figure 8-1. CLOCK GENERATOR TIMING

Diode D7, C15 and R18 make up the POC (power on clear) circuit. When power is applied, C15 starts to charge slowly until it reaches the threshold on pin 6 of U46, a Schmitt trigger. (By this time the logic and 5 volt supply have stabilized.) When the threshold is reached, pin 1 of U46 suddenly goes low. The resulting output on pin 8 of inverter U92 is initially low and then rapidly goes high. This signal is passed through a section of U77, a permanently enabled noninverting tri-state driver, as POC to S-100 Bus pin 99. It is also inverted in a section of U45 to become  $\overline{\text{POC}}$ .

The output on pin 8 of U92 is also connected to pin 15 of U63. Thus, pin 9 (RESET) of U63 is high to start the CPU in the reset condition when the Sol is initially turned on.

When  $\overline{\text{POC}}$  goes high, the RESET flip-flop section of U63 is free to clock. Assuming  $\overline{\text{PRESET}}$  is not active, it will change state on the first CLOCK transition. The resulting high on pins 10 and 5 of U63 cause pin 7 (READY) of U63 to go low to place the CPU in the not ready or wait state. This state is subsequently removed on the CLOCK transition following the transition which removed the low from pin 5 of U63. This helps prevent the CPU from starting in a crash condition.

The HOLD flip-flop (U64), however, is not affected by the POC circuit, and was clocked to a low on pin 7 well before the RESET and READY signals became active.

Operation of the POC circuit can also be initiated, without turning the power off, by a keyboard restart signal on pin 13 of J3 or by closing S1-1 if the N-P jumper is in. In either case, C15 is discharged through R58 and then allowed to recharge after  $\overline{\text{KBD RESTART}}$  is removed or S1-1 is opened.

$\overline{\text{POC}}$  also resets all stages of D flip-flop U76 (the phantom start-up circuit) to zero. On initial start-up, the CPU performs four fetch machine cycles (refer to Intel® 8080 Microcomputer Systems User's Manual) in accordance with program instructions. For each fetch, the CPU outputs a DBIN on pin 17. U76, connected as a four-stage shift register, is clocked by the inverted DBIN signal on pin 3 of NOR gate U46. Thus,  $\overline{\text{PHANTOM}}$ , on S-100 Bus pin 67, is active low (assuming the F-to-G jumper is in) for the first four fetches or machine cycles. After the fourth DBIN,  $\overline{\text{PHANTOM}}$  goes high.  $\overline{\text{PHANTOM}}$  is used to 1) disable any memory addressed in Page 0 that has Processor Technology's exclusive "Phantom Disable" feature and 2) cause the Sol program memory (ROM), which normally responds to Page C0 (hex) to respond to Page 00 (hex). The second function is discussed in Paragraph 8.5.2.



The inverted DBIN on pin 3 of U46 is also applied to pin 12 of NOR gate U46 and inverted to appear as PDBIN on S-100 Bus pin 78. This section of U46 also allows DIGI (bus pin 57) to override DBIN. (DIGI is used when an external DMA device replaces the CPU in terms of writing into and reading from memory.) The other CPU control signals (SYNC, INTE, HLDA, WR and WAIT) are also fed to the S-100 Bus pins as indicated. These, as well as DBIN or DIGI, are placed on the bus through tri-state drivers which are enabled by C/C DSB on S-100 Bus pin 19. Note that this signal is normally pulled high by R20.

The data lines of the CPU (D0-7) are bidirectional and are used for several functions. One of these is to output status at the start of each cycle which is marked by the SYNC output of the CPU. Status on D0-7 is latched in U93 and U106 (each of which contains four D flip-flops) when pin 8 of inverter U45 goes high. Status information, as identified on the schematic, is then buffered through tri-state drivers U94 and U107 to the S-100 Bus. The status latch strobe on pin 8 of U45 is extracted in the middle of the SYNC pulse by gating PSYNC and 02 in NAND gate U44. STAT DSB on S-100 Bus pin 18 is used to disable the U94 and U107 buffers when a DMA device or another processor assumes control of the S-100 Bus.

A second function of D0-7 is to output data from the CPU to the Bidirectional Data Bus. Data out of the CPU is placed on this bus through tri-state drivers (U80 and U81). Note that these drivers are normally enabled unless this bus is in the input mode or an external device has control of the bus. In the latter case, DO DSB on S-100 Bus pin 23 would be pulled low to make pin 8 of NOR gate U48 high. In the input mode pin 8 of U48 is high because OUT DSB is low. This signal is generated by decoding PAGE CC, MEM SEL, PORT IN FC, PORT IN FD, INT SEL to produce MPX ADR A and MPX ADR B on pins 3 and 11 respectively of two NOR gates in U48. MPX ADR A and MPX ADR B are decoded with DBIN on pin 5 of NAND gate U47.

The D0-7 bus lines are also used to input data to the CPU. Data input to the CPU is multiplexed from four data buses with four 4-to-1 line multiplexers (U65, 66, 70 and 79). These four buses are the: 1) Keyboard Data Bus, KDB0-7, 2) Parallel Input Data Bus, PID0-7, 3) Internal Data Bus, INT0-7, and 4) Bidirectional Data Bus, DIO0-7.

These data multiplexers are tri-state devices, with their outputs pulled up by R107 through R114 to a level that satisfies the input requirements of the CPU. Their outputs are active only when both their E1 and E2 (pins 1 and 15) are low. As can be seen, this occurs only when DBIN on pin 3 of NOR gate U46 is low; that is, when the DBIN output of the CPU is active to indicate its data bus is in the input mode.

Input selection to the multiplexers is done with the A and B inputs to U65, 66, 78 and 79. These two inputs are driven by MPX ADR A on pin 3 of NOR gate U48 and MPX ADR B on pin 11 of NOR gate U48. There are four possible states for the combination of MPX ADR A and B, and their relation to input selection is as follows:

1. If both are active (high), the multiplexers select the Bidirectional Data Bus.
2. When the keyboard is called up by the CPU, only PORT IN FC is active (low) to make MPX ADR A low. This selects the Keyboard Data Bus.
3. When the parallel port is called up by the CPU, only PORT IN FD is active (low) to make MPX ADR B low. This selects the Parallel Input Data Bus.
4. When the CPU selects any I/O port that uses the Internal Data Bus, only INT SEL (pin 2 of U47 and U61) is active. Thus, both MPX ADR A and B are low to select the Internal Data Bus.

Two other conditions, defined by PAGE CC on pin 2 and MEM SEL on pin 1 of NAND gate U44, are possible. When any of the four memory pages in the Sol are accessed, MEM SEL goes high and an inversion in U44 (PAGE CC is normally high) appears as a low MPX ADR A and B to select the Internal Data Bus. Should Page CC (the Display RAM) be addressed, PAGE CC also goes active (low) to override MEM SEL. MPX ADR A and B are consequently high to select the Bidirectional Data Bus. These two conditions are required since the ROM and System RAM use the Internal Data Bus and the Display RAM uses the Bidirectional Data Bus.

The address outputs of the CPU (A0-15) are placed on the Address Bus via tri-state drivers (U67, 68 and 81). These drivers are normally enabled since pin 3 of inverter U49 is pulled high by R36. ADD DSB on S-100 Bus pin 22 is used to disable the address drivers when a DMA device or another CPU takes over the bus.

A 5.1 volt zener diode, D11, and a divider network composed of R130, 131 and 132 derive -5 V dc from the -12 V dc supply for use by the CPU. Diode D12 and the same divider supply -12 V dc to pin 3 of U104, the phase clock conditioner.

#### 8.5.2 Memory and Decoder

Refer to the Memory and Decoder Schematic in Section X, Page X-16.

The System RAM consists of eight 1K by 1 bit static memory chips, U3 through U10, and it is assigned addresses C800-CBFF (hex). When the CPU wants to write data into memory, it addresses the System RAM on ADR0-15. ADR0-4 select the row inside the RAM chips, ADR5-9 select the column, and ADR10-15 select the page (in this case Page C8, hex). Page selection enables the eight RAM chips on pin 13. For a read operation, MWRITE on S-100 Bus pin 68 is low, and the resulting high on pin 3 (WE) of the RAM chips keeps them in the read mode. Thus, data on the Bidirectional Data Bus is read into the RAM's on their DI (pin 11) inputs. MWRITE is high, however, during the time the CPU wants to write data into memory. In this case, pin 3 of the RAM's is low to enable them to accept data from the Bidirectional Data Bus.

The ROM is also addressed on ADR0-15 as is the System RAM. Since there can be two pages, however, two enable lines (one for Page C0, hex, and the other for C4, hex) are provided. The C0 and C4 enables are connected to pins A6 and A5 respectively of J5, the Personality Module connector. Unlike the RAM, the ROM can only read data into the CPU, so the previously discussed MWRITE signal is not needed. Data out of the ROM is output on the Internal Data Bus on pins A3, A4 and B5-10 of J5.

ADR10-15 are input to the Address Page and Port Decoder (U34, 35, 36 and their associated logic). U34 (Address Page), U35 (Output Port) and U36 (Input Port) are 3-to-8 line decoders which have three enable inputs (G1, G2A and G2B). G1 must be high and both G2A and B must be low in order to obtain an active output.

Let's look at the Address Page Decoder, U34, first. It must be able to decode four pages: C0 and C4 (ROM), C8 (System RAM) and CC (Display RAM). (Note that these are the hexadecimal digits of the six high order address bits, ADR10-15).

The high order four bits (ADR12-15) must be 1100 (C, hex) in all cases by virtue of the U22 exclusive OR logic. If they are not, the G1 enable on U34 is low to disable that decoder. Bits ADR10 and 11 (The A and B inputs to U34) are the high order bits of the second hexadecimal digit which must be 00 (0, hex), 01 (4, hex), 10 (8, hex) or 11 (12, hex) if U34 is to have an active output. For C0, pin 11 of U34 is active (low); for C4, pin 10 is active; for C8 pin 9 is active; and for CC pin 7 is active. These outputs are applied to the appropriate memories and also provide the MEM SEL signal on pin 6 of one section in U23. (This section is actually a 4-input NAND gate which is functionally the same as a 4-input NOR gate.)

Note that the U22 logic input with ADR14 and 15 is also connected to PHANTOM. When this signal is active (low), the output on pins 3 and 11 will be low to disable U34 when ADR12-15 represent a C. If Page 0 is addressed, however, pins 3 and 11 of U22 are high, and this, coupled with lows on ADR10-13, are decoded by U34 as an active output on pin 11. The ROM will consequently respond to addresses in Page 0 and C0 (hex) as long as PHANTOM is active.

The other two enables on U34 (G2A and G2B) are connected to SINP and SOUT. These two status signals indicate an input or output operation during the CPU cycle. U34 is therefore disabled during these operations.

SINP and SOUT are also fed to pins 5 and 6 of NOR gate U53 which detects an input or output operation. Its output is inverted by U54 and applied to pin 9 of another U53 NOR gate. The other input (pin 8) to U53 is MEM SEL. So during a memory reference, input operation or output operation, pin 10 of U53 is active to enable the PRDY driver, U71. The low on pin 10 of U53 is also clocked by  $\phi 2$  as a high to pin 7 of U70, a J-K flip-flop that is connected as a D flip-flop. Note that the PSYNC  $\bullet \phi 2$  signal on pin 5 of U70 forces U70 to set during the middle of PSYNC (refer to CPU and Bus discussion). U70 cannot clock until pin 5 is released, and this occurs simultaneously with the low-to-high transition of  $\phi 2$ . PRDY is thus low immediately after pin 10 of U53 goes low and remains in that state from the middle of PSYNC to the first positive-going  $\phi 2$  after PSYNC. This is the time the CPU tests the status of the ready lines (PRDY and XRDY). If either is low, the CPU enters a WAIT state. U53, 70 and 71 thus guarantees that the CPU enters one WAIT state during cycles in which an input, output or memory reference is made.

U35 and 36, the Output and Input Port Decoders respectively, decode the higher order eight address bits (ADR8-15).

All Sol ports have a hexadecimal F (1111) in their high order four bits (ADR12-15 are 1's). The second hexadecimal digit is also never less than eight. This means that ADR11 is always 1 for a port address. These five address bits are thus NAND gated in U23 to provide one of the enables on U35 and 36. Note that the ADR14-15 combination is derived from the output on pins 3 and 11 of the U22 exclusive OR logic. This is permissible since no I/O operations are performed during the first four start-up cycles of the CPU.

The A, B, and C inputs to U35 and 36 (ADR8, 9 and 10 respectively) specify the second hexadecimal digit in the port address and are decoded to supply the indicated outputs. These outputs and their functions are defined in Table 8-1. U36 is enabled to decode when PDBIN and SINP are active; that is, during an input operation. U35 is enabled when SOUT and  $\overline{PWR}$  are active; that is, during an output operation.

INT SEL on pin 8 of inverter U83 is the remaining signal generated by the Input Port Decoder circuit. This signal is active when either input port F8, F9, FA or FB is decoded by U36.

Both the address page and input/output decoders can be disabled by SINTA (S-100 Bus pin 96) when the AE-to-AC and AB-to-AD jumpers are installed. SINTA is active (high) when the CPU is responding to an interrupt. Should an external device issue addresses during this time, any memory response would interfere with the

## PROCESSOR TECHNOLOGY CORPORATION

## Sol THEORY OF OPERATION

## SECTION VIII

Table 8-1. Port Decoder (U35 &amp; U36) Outputs and Their Functions.

PORT DECODER OUTPUT	FUNCTION
<u>PORT OUT FE</u>	Loads starting row address and first display line position information from Bidirectional Data Bus into Video Display scroll circuit.
<u>PORT OUT FD</u>	Clocks data from Bidirectional Data Bus to output data pins of PP connector.
<u>PORT OUT FB</u>	Loads data from Bidirectional Data Bus into Cassette Data UART.
<u>PORT OUT FA</u>	Clocks PP and CDI control bits from Bidirectional Data Bus.
<u>PORT OUT F9</u>	Loads data from Bidirectional Data Bus into SDI UART.
<u>PORT OUT F8</u>	Clocks RTS (request to send) from bit 4 of Bidirectional Data Bus to pin 4 of SDI connector.
<u>PORT IN FF</u>	Permits CPU to read data byte entered from Sense Switches.
<u>PORT IN FE</u>	Places Video Display scroll timer and screen position status on bits 0 and 1 of Bidirectional Data Bus.
<u>PORT IN FD</u>	Switches Data Input Multiplexer to input data pins of PP connector and resets PP at end of a transfer to ready it for another.
<u>PORT IN FC</u>	Switches Data Input Multiplexer to Keyboard Data Bus.
<u>PORT IN FB</u>	Strobes received data in CDI UART to Internal Data Bus.
<u>PORT IN FA</u>	Places PP, keyboard and CDI UART status on Internal Data Bus.
<u>PORT IN F9</u>	Strobes received data in SDI UART to Internal Data Bus.
<u>PORT IN F8</u>	Places SDI UART status on Internal Data Bus.

interrupt operation. To prevent this, SINTA is inverted in U58 to 1) disable U34 on pin 6 and 2) force pin 8 of NAND gate U23 high to disable U35 and U36 on pin 5. (This feature is provided to enable future versions of Sol to operate with a vectored interrupt system.)

### 8.5.3 Input/Output

Refer to the Input/Output Schematic In Section X, Page X-17.

This section in the Sol has five functional circuits: 1) Parallel I/O Logic, 2) Sense Switch Logic, 3) Keyboard Flag Logic, 4) SDI/UART and 5) Baud Rate Generator.

The PP uses U95 and 96 (4-bit D-type registers) and their related logic. Data output to the PP connector (J2) is latched from DIO $\phi$ -7 by U95 and U96. Data is strobed into these registers on the leading edge of an inverted active PORT OUT FD signal on pin 4 of inverter U54. This strobe is also applied to pin 2 of U73 which functions as a J-K flip-flop that is clocked by  $\phi 2$ . When the  $\phi 2$  goes from low to high 200 to 300 nsec after PORT OUT FD, pin 7 of U73 goes low to become POL on pin 17 of J2. (This delay allows U95 and 96 to stabilize.) U73 is reset in the middle of the following PSYNC which means POL is active for the balance of the cycle.

The outputs of U95 and 96 are tri-state outputs that are enabled by a low on pin 2. In the absence of POE at pin 15 of J2, pin 2 of U95 and 96 are low by virtue of the output on pin 8 of inverter U55. Note that the input to U55 is normally pulled up through R63. The POE provision permits tri-stating an external bidirectional data bus.

As discussed in Paragraph 8.5.1, parallel input data on J2 is fed directly to the Data Input Multiplexer (see Page X-15). The strobe that indicates the presence of input data, PDR on pin 4 of J2, is applied to pins 2 and 3 of one section in U72, a J- $\bar{K}$  flip-flop which is connected as a D flip-flop. When PDR goes active (low), pin 7 of U72 will go high on the next low-to-high transition of  $\phi 2$  to toggle the following U72 stage. At this point pins 9 and 10 of the second section in U72 go high and low respectively. Pin 9 supplies PIAK on pin 5 of J2. When high, PIAK signals the external device that Sol has yet to complete acceptance of the data. The state of pin 10 of U72 is transmitted to INT1 of the Internal Data Bus through a U71 tri-state noninverting buffer. U71 is enabled only for the duration of PORT IN FA (auxiliary status). During the time U71 is enabled, the CPU reads the Internal Data Bus. A high INT1 indicates the parallel input data is not ready; a low indicates the data is ready.

The second U72 flip-flop is preset by PORT IN FD or POC. PORT IN FD is active to read data in from the PP; POC occurs only when Sol is restarted or power is turned on. Thus the PP is reset and ready for another transfer at the end of a transfer or when POC is active.



PXDR on pin 16 of J2 is supplied by the external device. It indicates the device is ready to receive data.  $\overline{\text{PXDR}}$  is buffered to INT2 and will effect the transfer of data to the Internal Data Bus during the status input to the CPU.  $\overline{\text{PXDR}}$  is analogous to the previously discussed PIAK signal.

Sense Switches S2-1 through 8 are driven by  $\overline{\text{PORT IN FF}}$  when it is low. Thus, the DIO lines connected to closed switches are driven low, and those connected to open switches are pulled high.

U97 (a 4-bit D-type register) and one section of U52 (a J-K flip-flop connected as a D flip-flop) latch five bits of data on DIO3-7 when  $\overline{\text{PORT OUT FA}}$  goes active. These bits, which supply the indicated outputs, control conditions in both the PP and CDI. With respect to the PP, PIE enables parallel input, and PUS selects the parallel device for the transfer. The data in these two latches remains until either a new word is read out or POC goes active.

Also during  $\overline{\text{PORT OUT FA}}$ , the keyboard flag is reported.  $\overline{\text{KEYBOARD DATA READY}}$  on pin 3 of J3 is a low going pulse 1 to 10 usec in duration. It is applied to pin 13 of J-K flip-flop U70. Some time after pin 13 of U70 goes low, but before 500 nsec, U70 is set by  $\phi 2$  and pin 10 goes low. This low is buffered through U71 to INT $\emptyset$  to indicate the keyboard is ready to send data. Reset of U70 occurs with a POC or by  $\overline{\text{PORT IN FC}}$ . The latter occurs when data is accepted from the keyboard.

The other half of flip-flop U52, with its output on pin 6, latches one bit of status, DIO4, when  $\overline{\text{PORT OUT F8}}$  is active. Its output is applied to pin 5 of one operational amplifier section in U56 to become the SRTS (request to send) signal on pin 4 of J1, the SDI connector.

The SDI/UART centers around a UART, U51. The UART transmission conditions (parity, word length and stop bits) are determined by the settings of S4-1 through 5. (Refer to Paragraphs 7.5.8 through 7.5.10 in Section VII for descriptions of the switch settings and their effect on transmission.)

Data destined to leave Sol through the SDI/UART enters the UART on its T11-6 inputs from the Bidirectional Data Bus when TBRL (pin 23) is low; that is, when  $\overline{\text{PORT OUT F9}}$  goes active. Circuitry within the UART serializes the input data, which is in parallel form, and outputs it on pin 25 at a rate determined by the clock on pin 40. The binary states at pin 25 are low for a zero and high for a one. Assuming Sol is not in local operation ("off line"), the output on pin 25 of the UART is applied to pins 2 and 11 of J1 via two gates in U55 and the other half of U56.

Data that enters Sol through the SDI/UART on pins 3, 12 or 13 of J1 is input to the SDI UART on pin 20 by way of U38, an inverting level converter that converts data levels of up to  $\pm 25$  volts to TTL levels. (Note that current loop data on pin 12 or 13 of J1 is first rectified before it is applied to U38.) The UART converts this serial data into parallel form and outputs it on R01 through R08 (pins 12 through 5 respectively) to the Internal Data Bus when ROD (pin 4) is low; that is, when PORT IN F9 goes active.

The receive-transmit clock for the SDI UART is supplied by the Baud Rate Generator (U84, U85, U86 and their associated circuitry). U85 is a phase locked loop, U86 is a 7-stage binary counter and U84 is connected as a divide-by-11 counter. The 1200 Hz reference signal applied to pin 14 of U85 is supplied from the Video Display Generator. A phase comparator in U85 compares this signal to the output of a voltage controlled oscillator (VCO) in U85. By feeding an output from U86 (in this case the 1200 Hz output on pin 3) back to the compare input (pin 3) of U85, the circuit acts as a frequency multiplier. The output (pin 4) of U85 remains locked, therefore, to a multiple of its input on pin 14. In this case we have a 128X multiplier to generate 153.6 KHz which is counted down in U86. Since U86 is a 7-stage binary counter, the first stage output (pin 12) is 76.8 KHz (one-half of 153.6 KHz, the clock for U86), the second stage output (pin 11) is 38.4 KHz (one-fourth of 153.6 KHz), the third stage output (pin 9) is 19.2 KHz (one-eighth of 153.6 KHz), and so on to the seventh stage output (pin 3) which is 1.2 KHz (1/128 of 153.6 KHz).

With the exception of outputs on pins 12 and 9, the outputs of U86 are connected to S3, the Baud Rate Switch. The 19.2 KHz output on pin 9 is divided by 11 in U84 to supply 1745 Hz to S3-2. The 38.4 KHz on pin 12 can be connected to S3-8 instead of the 153.6 Hz clock by cutting the L-M connection and installing a jumper between K and M.

Let's now translate the frequencies input to S3 into Baud rates. The Baud rate of a UART is 1/16 of its clock rate. Thus, a 1200 Hz clock equates to a 75 Baud transmission rate, a 1745 Hz clock equates to a 109.1 (110) Baud rate, etc. It is now readily seen that the Baud rate available with S3-8 is 9600 assuming the L-M connection is made ( $153.6 \text{ KHz} \div 16 = 9600$ ). (The L-M connection is default wired on the Sol-PC; that is, there is a trace between L and M on the circuit board.) If the L-M trace is cut and a jumper is installed between K and M, the Baud rate with S3-8 is 4800 ( $76.8 \text{ KHz} \div 16 = 4800$ ).

We can thus select any one of eight clock frequencies for the SDI UART with S3, with the highest being determined by the K, L and M jumper arrangement. The selected clock is applied to both the receive and transmit clock inputs (pins 17 and 40 respectively) of the UART. This means, of course, that the UART always receives and transmits at the same Baud Rate.

Returning to the SDI UART, we see that its transmitter output on pin 25 is applied to pin 5 of U55, a two-input NAND gate that is functionally a NOR gate. It is normally enabled on pin 4 by pull-up resistor R44. A low on pin 5 represents a binary 0; a high represents a binary 1. The inverted output on pin 6 of U55 is again inverted (assuming Sol is not operating in Local) by the following U55 NAND gate. One-half of operational amplifier U56, operating open loop, converts TTL levels to RS-232 levels (5 to 15 volts). Pin 3 of U56 is held at +2.5 V dc by the R47 and R48 divider network. When pin 2 is more negative than pin 3, the output on pin 1 of U56, which is fed to pin 2 of J1, is at approximately +10 volts. For the opposite condition, pin 2 of J1 is about -10 volts. Thus, U56 also inverts, and a high or low on pin 2 of J1 represent a binary 1 and 0 respectively.

Two conditions can override transmitted data: a keyboard break (BRK) or local (KBD LOC) command. For a break command, BRK on pin 4 of J3 and pin 4 of NOR gate U55, is low to hold pin 6 of U55 high for the duration of the BRK signal. This appears as a "space", or high level, on pin 2 of J1. (A space, or break, condition requires that the space level exist for a period longer than the normal length of a character.) In the case of a KBD LOC command from the keyboard, pins 1 and 13 of the other two U55 sections are low. Thus, data cannot be transmitted to pin 3 of NAND gate U55, and pin 11 of NOR gate U55 is held high to enable tri-state driver U37 at pin 15. Data on pin 6 of U55 is consequently looped back by way of U37 and R21 to pin 12 of U38. Data on pin 12 of U38 overrides any data arriving at pin 13 of U38. In local operation, therefore, data from pin 25 of the UART does not appear at pin 2 of J1, but it is looped back to the receiver input (pin 20) of the UART via U37, R21 and U38.

Notice that data on pin 25 of the UART will also be looped back if S4-6 is closed (half duplex operation). But in this case, data from the UART is also fed to pin 2 of J1.

Serial data from the UART that appears at pin 1 of U56 also drives transistor Q1 by way of R45 and R46 to supply the serial current loop output (SCLO) on pin 11 of J1. Q1 supplies 20 ma. (max.) current for a binary 1 and no current for a binary 0.

Pin 23 of J1 (connected through R23 to +12 V dc) is the serial loop current source (SLCS). It can supply up to 20 ma of current to ground and is used when the external current loop device has no current source.

Data received from a current loop device enters Sol on pins 12 and 13 of J1 in the form of no current for a 0 and 20 ma of current for a 1. This input is rectified by bridge rectifier D3-D6 and applied to a light emitting diode (LED) in optical isolator U39. As its name implies, U39 electrically isolates the current loop circuit from the rest of the Sol. (This isolation permits a high offset voltage on pins 12 and 13 of J1.) For a 1, the LED is energized, and

the light is optically coupled to the base of a photo transistor in U39 to cause the transistor to conduct. Conduction translates to a low, or mark, level at the input (pin 13) of U38. Since both the current loop and RS-232 received data (SLR1/SLR2 and SRD respectively) share the input to U38, both should not be used simultaneously.

There are five external control signals in the RS-232 section of the SDI/UART: two are sent to the external device (SRTS and SDTR), and three are received from the device (SCTS, SCD and SDSR).

SRTS on pin 4 of J1 was discussed earlier. SDTR (serial data terminal ready) is simply tied to +12 V dc through R24. This indicates to the external device that Sol is connected to it.

SCTS (serial clear to send), SCD (serial carrier detect) and SDSR (serial data set ready) indicate status of the external device. They enter Sol on pins 5, 8 and 6 of J1 respectively, and all three are active high. Following level conversion and inversion in line receivers U38, data on these lines is gated through noninverting tri-state buffers U37 to the Internal Data Bus when PORT IN F8 is active.

PORT IN F8 also enables five bits of UART status to be reported over the Internal Data Bus. These are PE, FE, OE, DR and TBRE on pins 13, 14, 15, 19 and 22 respectively of the UART. They are defined as follows:

- PE: Parity Error--received parity does not compare to that programmed. (Bit INT2)
- FE: Framing Error--valid stop bit not received when expected. (Bit INT3)
- OE: Overrun Error--CPU did not accept data before it was replaced with additional data. (Bit INT4)
- DR: Data Ready--data received by UART is available when requested. (Bit INT6)
- TBRE: Transmitter Buffer Register Empty--UART is ready to accept another word from the Bidirectional Data Bus. (Bit INT7)

#### 8.5.4 Display Section

An understanding of how characters are formed on the video monitor will help you follow operation of the display section.

The monitor screen can be thought of as a large matrix of small light elements, or dots, that can be turned on and off. In this context the overall video presentation consists of light and dark dots.

In the Sol, the display format is 64 characters maximum per character row, with a maximum of 16 rows per frame (page). Thus, up to 1024 characters can be displayed per page.

A 9 x 13 (columns by lines) dot area, or character position, is allotted on the monitor screen for each displayed character (see Figures 8-2 and 8-3 on Page VIII-24). Consequently, each character row consisting of sixty-four 9 x 13 dot areas requires 13 horizontal scan lines. To provide spacing between both characters and rows, only 12 dot lines and seven dot columns within the 9 x 13 matrix are used for character display. Only nine of the available 12 dot lines, however, are used for any given character.

Let's take a closer look at how the 9 x 13 dot matrix is used. The first seven dot columns are available for all character displays; the last two are used to provide a space between characters. The first dot line in a character row is always blank to provide a space between character rows. As shown in Figure 8-2, the second through tenth dot lines are available for all upper case (capital) and control characters, all symbol and punctuation marks (except the comma and semicolon), and all lower case characters (except the g, j, p, q and y). As shown in Figure 8-3, dot lines five through 13 are available to display characters that normally extend below the base line--lower case g, j, p, q and y plus the comma and semicolon.

Now that we have a feeling for how characters are formed on the video monitor screen, we will move on to the circuit description.

Refer to Display Section Schematic in Section X, Page X-18.

The 14.31818 MHz DOT CLOCK, which defines the period of one dot (69.8 nsec) in a character display matrix, controls all timing in the Video Display Generator. DOT CLOCK is applied to pin 2 of U28, a four-bit binary counter that is preset to count from seven through 15 to divide DOT CLOCK by nine. Two 1.591 MHz outputs are supplied by U28: LOAD CLOCK on pin 11 and CHARACTER CLOCK on pin 12. Pin 11 is a low-active pulse of one DOT CLOCK duration. Pin 12 is high for five and low for four DOT CLOCK periods. Both the LOAD and CHARACTER CLOCK low-to-high transitions occur synchronously on the same DOT CLOCK.

CHARACTER CLOCK, which defines the period of one character position (628 nsec), is inverted in U49 to become CHARACTER CLOCK. It performs most of the clocking functions in the Video Display Generator and is made available on pin 4 of J4 for use by external graphic display devices.

CHARACTER CLOCK is in turn divided in U31 and U33, both of which are presettable four-bit binary counters. Both start at count 3 when pin 8 of NAND gate U47 is low, and together they count 102 CHARACTER CLOCKS to define horizontal timing at 64 usec (102 x 628 nsec = 64 usec).

CHARACTER ADDRESS*	LINE ADDRESS	SCAN LINE NO.	COLUMN NO.									VIDEO INFORMATION BITS
			1	2	3	4	5	6	7	8	9	
1001001 ↑ ↓ 1001001	1111	1	0	0	0	0	0	0	0	0	0	000000000 (blank)
	0000	2	0	●	●	●	●	●	0	0	0	011111000
	0001	3	0	0	0	●	0	0	0	0	0	000100000
	0010	4	0	0	0	●	0	0	0	0	0	000100000
	0011	5	0	0	0	●	0	0	0	0	0	000100000
	0100	6	0	0	0	●	0	0	0	0	0	000100000
	0101	7	0	0	0	●	0	0	0	0	0	000100000
	0110	8	0	0	0	●	0	0	0	0	0	000100000
	0111	9	0	0	0	●	0	0	0	0	0	000100000
	1000	10	0	●	●	●	●	●	0	0	0	011111000
	1001	11	0	0	0	0	0	0	0	0	0	000000000 (blank)
	1010	12	0	0	0	0	0	0	0	0	0	000000000 (blank)
	1011	13	0	0	0	0	0	0	0	0	0	000000000 (blank)

\*7-bit ASCII code for I

● = illuminated dot

Figure 8-2. Example of uppercase character (I) display.

CHARACTER ADDRESS*	LINE ADDRESS	SCAN LINE NO.	COLUMN NO.									VIDEO INFORMATION BITS
			1	2	3	4	5	6	7	8	9	
1110000 ↑ ↓ 1110000	1111	1	0	0	0	0	0	0	0	0	0	000000000 (blank)
	0000	2	0	0	0	0	0	0	0	0	0	000000000 (blank)
	0001	3	0	0	0	0	0	0	0	0	0	000000000 (blank)
	0010	4	0	0	0	0	0	0	0	0	0	000000000 (blank)
	0011	5	●	0	●	●	●	0	0	0	0	101110000
	0100	6	●	●	0	0	0	●	0	0	0	110001000
	0101	7	●	0	0	0	0	●	0	0	0	100001000
	0110	8	●	0	0	0	0	●	0	0	0	100001000
	0111	9	●	●	0	0	0	●	0	0	0	110001000
	1000	10	●	0	●	●	●	0	0	0	0	101110000
	1001	11	●	0	0	0	0	0	0	0	0	100000000
	1010	12	●	0	0	0	0	0	0	0	0	100000000
	1011	13	●	0	0	0	0	0	0	0	0	100000000

\*7-bit ASCII code for p

● = illuminated dot

Figure 8-3. Example of lowercase character (p) display.



As indicated in Figure 8-4 on Page VIII-27, Subgroup Counter U31 and Group Counter U33 are preset to a count of 3 at the start of each horizontal scan line. U31 counts from 3 through 15 (13 character positions) and enables U33 for one count. U31 then counts 0 through 15 and enables U33 for the second count. The sequence continues through four more groups of 16 character positions, and at this point U33 is at its sixth count (a binary 9). Thus, pins 11 and 14 are high at pins 10 and 11 of U47. U31 continues to count from 0, and on the ninth count (a binary 8) pin 9 of U47 goes high. The resulting low on output pin 8 of U47 loads three into U31 and U33, and the cycle repeats. The U31-U33 cycle, from preset, is then 13, 16, 16, 16, 16 and 9 character position counts for a total of 102.

The QD output on pin 11 of U33 is SCAN ADV, and the QC output on pin 12 is HDISP. SCAN ADV is used to generate horizontal synchronization signals, and HDISP defines the start of the display portion of the horizontal scan line.

Four outputs from U31 and the two low order outputs of U33 (pins 13 and 14) are input to the Character Address Multiplexer, U30 and U32, which supplies the low order six address bits to the Display RAM (U14 through U21). The second address source for the Display RAM is the Address Bus, bits ADR0-5. Address source selection is controlled by the output on pin 7 of D flip-flop U75. Pin 7 of U75 goes high when PAGE CC (the Display RAM) is active and PSYNC • 02 goes high (which it does in the middle of PSYNC). Pin 7 of U75 remains high for the rest of the memory access cycle.

The preset signal (pin 8 of U47) to U31 and U33 is applied to the Scan Counter (U40) via inverter U87. U40 counts the horizontal scan lines that make up a row of characters and supplies the line number to U25, the Character Generator ROM. (This ROM is discussed later.) U40 is preset to a count of 15 for the first scan line in the character row. It then counts from 0 through 11. On count 11, SCAN ENABLE on pin 8 of U47 is inverted in U87 to disable the Scan Counter. A decoder, comprised of NAND gates U59 and U60, decodes the 13th count (count 11) in U40 and SCAN ENABLE to supply a load pulse to pin 9 of U40. This resets U40 to a count of 15, and the cycle repeats. (Presetting the Scan Counter to a count of 15 permits the Character Generator ROM to provide a blank spacer line between character rows since line 15 in the ROM is always blank.)

The output on pin 8 of NAND gate U59, after inversion in U87, becomes the OVERFLOW LINE signal. This signal occurs after each character row and appears at pins 7 and 10 of Text Counter U62 to enable it to count. Thus, the Text Counter counts character rows. It resets itself with its carry output (pin 15) through another inverter in U87, with the reset count being determined by the state on pin 10 (VDISP) of J-K flip-flop U43. If VDISP is low, the Text Counter resets to a count of 0; if VDISP is high, it resets to a count of 12.

Assume VDISP is active (low), which it is during the vertical display portion of the displayable area on the screen. (Refer to Figure 8-4.) U62 is then preset to a count of 0 and will count from 0 through 15 (16 character rows). The resulting carry output on count 15 of the Text Counter causes the U43 VDISP flip-flop to toggle. It also appears as a low on the load input of the Text Counter. The Text Counter is also enabled to reset by virtue of the OVERFLOW LINE going low after the reset of the Scan Counter. Since VDISP is now high, the Text Counter is reset to a count of 12 and will count 12 through 15 (four character rows). The carry output from the Text Counter then causes the U43 VDISP flip-flop to toggle, and the Text Counter is reset to a count of 0. We can now see that the Text Counter counts 16 character rows when the display is active (VDISP is low) and four character rows when the display is blanked (VDISP is high). The total of 20 character rows represents a full display of 260 scan lines for 60 Hz operation (13 scan lines/row x 20 rows = 260 scan lines per page).

Horizontal and vertical synchronization signals are generated by two one-shot multivibrators consisting of three two-input NOR gates in U102. Horizontal sync is triggered by SCAN ADVANCE and vertical sync by VDISP. Both circuits generate fixed-length sync pulses with adjustable starting times. C52 determines the length of the horizontal sync pulse and C53 the length of the vertical sync pulse. The starting times, with respect to triggering, are variable with variable resistors VR1 (HORIZ) and VR2 (VERT) to provide continuous adjustment of the display position on the screen. An exclusive OR gate in U74 combines the two sync pulses into a composite sync (COMP SYNC) signal. Note that the use of the exclusive OR inverts the horizontal sync pulses when the vertical sync pulse appears. Since vertical sync information is extracted in a monitor by an integrating, or averaging, process, this technique maintains horizontal synchronization during the vertical sync period.

Two types of blanking are available: control character blanking and video blanking. The first blanks control characters and causes cursor information to be displayed in their place. Video blanking forces portions of the video display to a white or black level, depending on whether normal or reverse video is selected with S1-4.

Control character blanking, switch selectable with S1-3, is accomplished with one NAND gate in U60 and one NAND gate in U61. When a control character is present in the Data Latch (U26 and U27), pins 3 and 15 of U26 are high. Assuming the blanking option is selected (S1-3 closed), the output of U60 (LOAD CLOCK) is gated with the control character bits by U61 to clear the video parallel-to-serial converter, U41. U41 then loads all zeros instead of the character.

Video blanking is initiated by the PRE BLANK or COMP BLANK (pin 14 of Blank Latch U42) inputs to U59, a three-input NOR gate. The third input, the video output on pin 6 of exclusive OR gate U74, is blanked when any of the two blanking inputs is active.

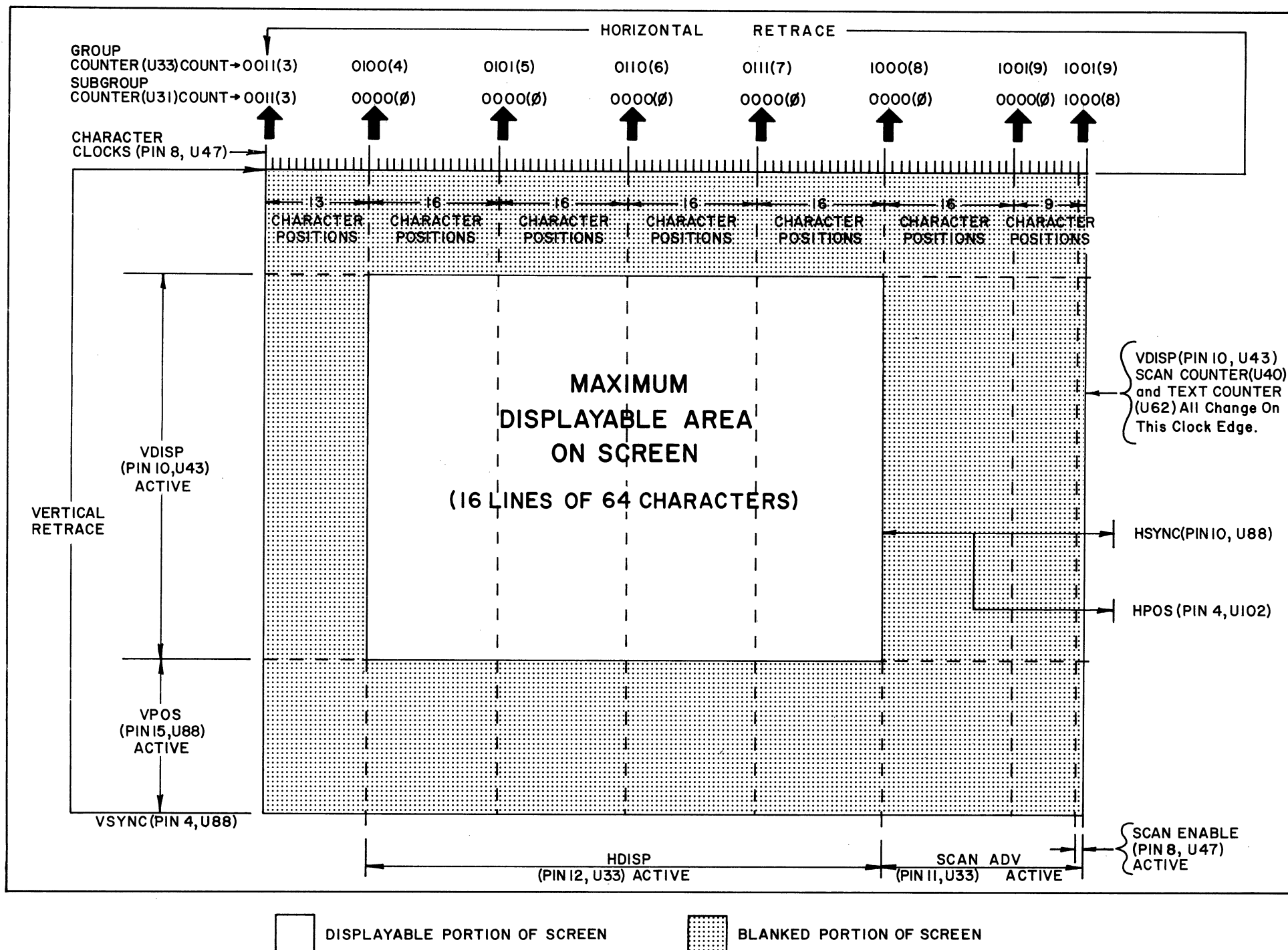


Figure 8-4. VIDEO DISPLAY TIMING

The PRE BLANK input provides "window shade" blanking which is analogous to pulling a window shade down from the top of the display. PRE BLANK is generated in one half of J-K flip-flop U43. U43 is reset by the TC output of First Screen Position Counter, U11, and set by VDISP. The output on pin 7 of U11 is generated by the scrolling circuitry (to be discussed later) and defines the character row for which the "window shade" ends. It may begin with any character row from zero through 14.

The remaining video blanking function concerns the output on pin 14 of D flip-flop U42. This signal, COMP BLANK, is a composite of HDISP and VDISP.

Since there is a two character time delay between Display RAM addressing and the corresponding video output on pin 6 of exclusive OR gate U74, the horizontal and vertical blanking signals must be delayed an equal amount. U42, connected as a two-stage shift register, functions to shift the blanking into synchronization with the video. Since U42 is clocked by LOAD CLOCK (which has a period equal to one character time), COMP BLANK is delayed two character times from the input on pin 4 of U42. COMP BLANK is active low during nondisplayable portions of the video scan to override any video input data on pins 1 and 2 of NOR gate U59. The display is thus blanked.

The Display RAM consists of eight 1K x 1 bit RAM (random access memory) chips, U14 through U28. All chips are held permanently enabled by connecting their CE (pin 13) inputs to ground. Memory addressing is provided through two-to-one multiplexers (U30, U32 and U12) which select one of two display address sources: 1) an external address on Address Bus bits ADR0-9 and 2) an internal address supplied by the Subgroup Counter (U31), Group Counter (U33) and the Beginning Address Counter (U1). The function of the address bits associated with each address source is as follows:

1. External address bits ADR0-5 specify the character position (one of 64) in the character row.
2. External address bits ADR6-9 specify the character row position (one of 16) on the display screen.
3. Internal address bits, a total of six outputs from U31 and U33, specify the character position (one of 64) in the character row.
4. Internal address bits, the four outputs from U1, specify the character row position (one of 16) on the display screen.

Normally the internal display address is multiplexed to the Display RAM. When the CPU or a DMA device requests access (PAGE CC active), the multiplexers switch to the external address lines, ADR0-9.

Seven-bit ASCII-coded data is written into RAM chips U14 through U20 from bits DIO0-6 of the Bidirectional Data Bus, and the cursor bit (DIO7) is written into RAM chip U21. This writing occurs when the write enable (WE) input to the RAM chips is low. This occurs when the Display RAM is addressed (PAGE CC active low) and MWRITE on S-100 Bus pin 68 is high. The enable is supplied on output pin 8 of NAND gate U44. Data is read out of the Display RAM when pin 8 of U44 is high. Data out of the Display RAM is placed on the Bidirectional Data Bus via tri-state drivers U29 and U89 when PAGE CC and PDBIN (S-100 Bus pin 78) are active. U29 and U89 are enabled by a low output on pin 11 of another U44 NAND gate.

Data out of the Display RAM is also strobed into Data Latches U26 and U27 by LOAD CLOCK. Seven outputs from these latches are used to address the Character Generator ROM, U25. Note that the output from RAM chip U19 is inverted in exclusive OR gate U74 before being applied to the C input (pin 13) of U26, and the complement (pin 14) of the QC output of U26 is used in addressing U25. This is done so that the Data latches will output the space code (01000000) to the Character Generator ROM when the latches are reset. These latches are reset each time PAGE CC is active by way of U75, a J-K flip-flop connected as a D flip-flop, and D flip-flop U42 ( $\bar{Q}$  output pin 6). By outputting the space code on reset, the Data Latches insure a blank character position on the screen.

The Character Generator ROM, U25, has seven character address inputs (A1 through A7), four scan line inputs (RS1 through RS4) and seven data outputs (B1 through B7). It is programmed to generate seven bits (dots) of character information for the selected scan line of the character row. U25 also automatically blanks scan lines that are not a part of the character and shifts the g, j, p, q, y, comma and semicolon to the fifth through 13th scan lines in the dot matrix (refer to Figures 8-2 and 8-3 on Page VIII-24). Complete patterns for the 6574 and 6575 Character Generator ROM's are provided in Figures 8-5 and 8-6 respectively. Note that the address bits A0 through A6 in Figures 8-4 and 8-5 correspond to the A1 through A7 inputs to U25 on the schematic, scan lines R0 through R8 are specified by the RS1 through RS4 inputs to U25 on the schematic, and the data output bits D0 through D6 correspond to the B1 through B7 outputs from U25 on the schematic.

Let's see how the Character Generator ROM produces a character using an uppercase "C" and "T" as an example. In this example, these two characters are to be displayed in the first and second character positions respectively on the third character row of the display screen. Remember that the character position and row parameters are contained in the Display RAM since the 7-bit ASCII-coded

A3..A0		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
A6..A4		D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0
000	R0																
	R8																
001	R0																
	R8																
010	R0																
	R8																
011	R0																
	R8																
100	R0																
	R8																
101	R0																
	R8																
110	R0																
	R8																
111	R0																
	R8																

= Shifted character. The character is shifted three rows to R3 at the top of the font and R11 at the bottom.

Figure 8-5. 6574 Character Generator ROM pattern.

"C" and "T" were stored in the RAM in the proper character positions in the third character row.

After the first two character rows have been displayed, the Scan Counter (U40) is reset to a binary count of 15 (1111) and the Character and Line Address Multiplexers (U30, U32 and U12) call up the "C" in the Display RAM. The Scan Counter output specifies line 15 in the Character Generator ROM on RS1 through RS4. As previously mentioned, this line in the ROM is blank. Thus, the first scan line of the third character row is blank.

The 7-bit ASCII code for the "C" (1000011) is input from the Display RAM to address the Character Generator ROM by way of the Data Latches (U26 and U27). This address is applied to ROM inputs A7 through A1 (A6 through A0 in Figures 8-5 and 8-6). The Scan Counter changes to a count of zero which specifies scan line R0 in the Character Generator ROM. As shown in Figures 8-5 and 8-6, the ROM in turn outputs a 7-bit word, 0011110, on D6 through D0 respectively (B7 through B1 on the schematic).



A3..A0		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
A6..A4		D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0	D6...D0
000	R0																
	R8																
001	R0																
	R8																
010	R0																
	R8																
011	R0																
	R8																
100	R0																
	R8																
101	R0																
	R8																
110	R0																
	R8																
111	R0																
	R8																

= Shifted character. The character is shifted three rows to R3 at the top of the font and R11 at the bottom.

Figure 8-6. 6575 Character Generator ROM pattern.

For the second character position the Character and Line Address Multiplexers call up the "T" in the Display RAM. The resulting ASCII code for a "T" (1010100) ultimately appears on the address inputs to the Character Generator ROM. Since the Scan Counter is still at a count of zero, the ROM outputs 1111111. This process continues for the balance of the displayable portion of the video scan line.

At the end of the horizontal scan line, the Scan Counter changes to a binary count of 0001 which specifies scan line R1 in the Character Generator ROM. The "C" and "T" are again called up from the Display RAM for the first and second character position respectively. The ROM consequently outputs 01000001 and then 00010000. This sequence continues through scan line R8 when the Scan Counter is at a count of 8 (1000) to produce the "C" and "T".

As discussed earlier, the Scan Counter cycles through 13 counts or scan lines. For the "C" and "T" in our example, the Scan Counter has counted ten lines (15, 0, 1, 2, 3, 4, 5, 6, 7 and 8). The remaining three scan lines are not used in forming the "C" or "T", so on counts 9, 10 and 11 of the Scan Counter the Character

Generator ROM automatically outputs all zeros for these two character positions. After the last scan line in the third character row, the Scan Counter is reset to a count of 15 to start the fourth character row.

The Character Generator ROM output is converted from parallel to serial form in an 8-bit shift register (U41) that is clocked by DOT CLOCK. For each high bit on the input, the serial output (QH, pin 13) of U41 is high for one DOT CLOCK period. For each low bit, QH is low for one DOT CLOCK period. Note that parallel input bit PH (pin 14) is tied to ground. This effectively adds a low bit (or dot) following the data and provides one of the spacer dots between characters. The second spacer dot is generated by connecting the serial input (pin 1) to ground and applying LOAD CLOCK to the load (LD, pin 15) input to U41. When LOAD CLOCK goes low, which it does every ninth DOT CLOCK, U41 shifts in one zero.

A blink oscillator (two inverter sections in U88), a latch (one section in U42) and their associated components comprise the cursor circuit. The blink oscillator runs continuously at a rate set by R84 and C36. Its output has a nominal 0.5 sec period. If the blink option is selected with S1-5, the blink signal is applied to one input of a gate in U60. The other input to this gate is provided by the blink latch, one section in U41. If the cursor bit QA out of Data Latch U26 is high, D flip-flop U42 sets for the time the ROM is active on the character and remains set during the period when video data is shifted out of U41. The output of U42 is gated high through NAND gate U60 when BLINK (pin 6 of U88) is low. BLINK is held low when the blink option is not selected. The output of U60 is in turn gated with the video output of U41 in U74, an exclusive OR gate. U74 thus inverts the video if the output of U60 is high, and no inversion takes place if the output of U60 is low.

The video signal including the cursor, is gated to pin 9 of another U74 exclusive OR gate in the absence of any blanking signals at the other two inputs to NOR gate U59. If S1-4 is open, U74 inverts the video signal to produce a reverse (black on white) display. Raw video on pin 8 of U74 is supplied to pin 15 of J4. Video out on pin 6 of inverter U87 is combined with COMP SYNC on pin 8 of another U87 inverter in a resistive mixer, R80-R82, to meet EIA composite video signal standards, and coupled to P1 for use by a video monitor. This mixer has a 61-ohm output impedance.

Both Beginning Address Counter U1 and First Screen Position Counter U11 are enabled to advance their counts when pin 9 of J-K flip-flop U75 is low, which it is for about 600 nsec following OVERFLOW LINE; that is, after the Scan Counter (U40) is loaded. This, of course, occurs at the end of every scan line in the character row.

The scroll circuit consists of U1, U11, Scroll Control Latch U2 and Screen Position Control Latch U13 and their associated circuitry. U1 and U11 are up and down counters respectively that are pre-

set to the outputs of latches U2 and U13. U2 latches the starting row address from DIO0-3 and U13 latches the data on DIO4-7, with PORT OUT FE being the strobe. Data on DIO4-7 specifies where the first line will be displayed. Thus, the number loaded into U1 is the address of the first displayable scan line, and the number loaded into U11 defines the character row (0 through 15).

U11 is preset by VDISP from pin 9 of J-K flip-flop U43. This means U11 is forced to its preset condition from the end of the displayed text to the top of the next character row. During this time, pin 6 of another U43 J-K flip-flop is set high to preset U1. If U11 is preset to 0, its TC output on pin 7 is low and pin 6 of U43 is reset to a low. This allows U1 to count with each horizontal scan line.

If U11 is preset to any number other than 0, pin 6 of U43 cannot be reset low until U11 reaches zero. Assume U11 is preset to two. It must count down two character rows before U1 starts counting. During this time, pin 7 of U43 (PRE BLANK) is low, and as previously discussed, the display is blanked.

We can now see that the PRE BLANK time, often called "window shade", is variable with the number loaded into U11. Therefore, scrolling is performed by changing the numbers in U2 and U13 without the need to reposition the text within the Display RAM.

The remaining circuit in the Display Section consists of transistor Q2, one section of U87, 89 and U102. U88 and U102 are connected as a one-shot 250 msec timer that is triggered when PORT OUT FE goes active (pin 1 of inverter U87 goes high). Thus, when data is loaded into U2 and U13, this timer starts. Tri-state driver U89, which is enabled by PORT IN FE, transmits the state of this timer to DIO0 on the Bidirectional Data Bus. The CPU can consequently test the timer status by looking for a high on DIO0. This timing allows a 250 msec scroll rate without the need for complex timing routines in the CPU. Q2, R102 and C37 serve to speed up timer reset.

#### 8.5.5 Audio Tape I/O

Refer to Audio Tape I/O Schematic in Section X, Page X-19.

Timing for the Audio Tape I/O is derived from the 1200, 2400, 4800, 19,200 and 38,400 Hz signals received from the Baud Rate Generator in the Input/Output section of Sol. The first two are used by the write data synchronizer (U100) and the digital-to-audio converter (U101).

The remaining three signals are fed to two sections of U111, a quad multiplexer or select gate. All four sections of U111 are used to select clocks for low speed or high speed operation according to the select inputs, pins 9 (A) and 14 (B). The states of these two select inputs must be complementary to each other in order to select

the high or low speed clocks. Specifically, A must be high and B low to select high speed clocks; the converse condition selects low speed clocks. The select inputs are supplied by TAPE HI SPEED and TAPE HI SPEED.

The output of the second section on pin 11 of U111 is BYTE WRITE CLOCK, 4800 Hz on low speed and 19.2 KHz on high speed. The third section outputs a 19.2 KHz (high speed) or 38.4 KHz (low speed) timing signal to input pin 10 of binary up counter (U112).

RECOVER CLOCK is produced by a phase locked loop (U110), another U112 binary up counter and the first and fourth sections of U111. The signal input (pin 14) to U110 is supplied from output pin 1 of D flip-flop U113. It is a constant frequency, regardless of whether one or two transitions are detected in the read data during the count out time (12 counts) of the U112 counter with outputs on pins 13 and 14. A phase comparator in U110 compares the signal input to the output of a voltage controlled oscillator (VCO) in U110 (pin 4). By feeding the VCO output through a counter (the other half of U112) before feeding the counter output back to the compare input (pin 3) of U110, the circuit acts as a frequency multiplier. The output of this circuit remains locked, therefore, to a multiple of the signal input on pin 14 of U110.

The output of U110 is nominally 19.2 KHz. The actual output is determined by the signal input which in turn is a function of tape speed. In other words, the phase lock loop circuit tracks input frequency variations. And it will track such variations within its locking range which is determined by the setting of variable resistor VR3 (connected to pin 12 of U110).

For high speed, the divide-by-four output of U112 (pin 4) is selected as RECOVER CLOCK. For low speed, the VCO output of U110 is selected for RECOVER CLOCK. This clock serves as read clock for the CDI UART, U69.

CDI control involves PORT IN FA, PORT IN FB, PORT OUT FB, TAPE CONTROL 1 and 2, POC (power on clear), TAPE HIGH SPEED and TAPE HI SPEED. The last two were previously explained in the discussion of U111. PORT IN FA strobes the CDI UART status (DR, TBRE, OE and FE--refer to Page VIII-22 for definitions) to the Internal Data Bus, INT3-7. PORT IN FB strobes received data on pins 5-12 of U69 to the Internal Data Bus, INT0-7. PORT OUT FB loads data from the Bidirectional Data Bus (DIO0-7) into U69. POC simply resets U69 whenever power is applied to the Sol.

TAPE CONTROL 1 and 2 are used to turn one or two recorder motors on and off. An active low TAPE CONTROL 1 energizes K1 to close its contacts and turn recorder #1 on; a high de-energizes K1 to turn the recorder off. TAPE CONTROL 2 does the same thing with K2 to control another recorder. Diodes D13 and 14, which shunt K1 and K2

respectively, prevent damage to the logic circuitry in the Input/Output section due to inductive kickback. R155 and 156 are current limiters that keep the relay contacts from "welding" together.

When the CDI is in the write mode, data is input to the UART (U69) under control of ~~PORT OUT FB~~. Upon completion of this strobe, the transmit sequence is initiated within the UART, with the transmission rate being governed by BYTE WRITE CLOCK.

The transmission sequence begins with a start bit, a low (data zero) on the UART's TO output. It is followed by eight data bits and two stop bits (high on the UART's TO output), with the number of bits being fixed by the connections to pins 34 through 39 of U69.

The data from U69 is applied to the D input of D flip-flop U100 which is clocked at 1200 Hz. Consequently, the output on pin 1 of U100 follows the input data on pin 5 after the rising edge of the 1200 Hz clock. This output is connected to the reset (pin 4) of U101, so when the data out of the UART is high, the first section in U101 is forced to a reset condition. In this condition the J and K inputs to the second stage of U101 are held high which allows the flip-flop to change state on the rising edge of the clock.

The clock for U101 (OUTPUT CLOCK) is 2400 Hz in the high speed mode or 4800 Hz in the low speed mode. This clock is derived from 2400 Hz in conjunction with the low speed select signal in NAND gate U98 and exclusive-OR gate U99.

In the high speed mode, pins 12 and 13 of U98 are held low, thus holding pin 10 of U98 high. As a result the 2400 Hz signal is inverted in U99 to become the clock for U101.

Pins 12 and 13 of U98 are held high, however, in the low speed mode to enable U98. In this case R117 and C47 provide a delay in the U98 gate. When the 2400 Hz signal on pin 2 of U99 changes state, so does pin 3 of U99. Also, C47 charges through R117 for several usec, at which point pin 10 of U98 is brought to the opposite polarity. The output from U99 then goes high. A series of positive pulses, with a pulse width approximately equal to the R117, C47 time constant (10 usec) and occurring at every transition of the 2400 Hz signal, appears on pin 3 of U99. This circuit thus operates as a frequency doubler in the low speed mode to provide a 4800 Hz clock for U101.

The 2400 Hz signal from which the U101 clocks are derived also produces the 1200 Hz clock signal for U100. As a result the 1200 Hz signal changes state following a propagation delay after the 2400 Hz signal falls.

As previously stated, the second stage of U101 is allowed to change state on the positive going transitions of the OUTPUT CLOCK as long as the data out of the synchronizer is a "1". The end result is an output on pin 14 of U101 that is one-half the clock frequency (1200 Hz and 2400 Hz in the high and low speed modes respectively).

Assume the data stream out of the UART goes low ("0"). On the next rising edge of the 1200 Hz signal, U100 will reset with Q low and  $\bar{Q}$  high. A low reset on pin 4 of U101 enables the first U101 stage to toggle on the next rising edge of the OUTPUT CLOCK which occurs 1/2400 second after the synchronizer output falls. Remember that OUTPUT CLOCK moves from a low to a high shortly before the 1200 Hz signal did. The reset on pin 4 of U101 is thus removed slightly after the OUTPUT CLOCK occurred. With the J and K inputs to the first U101 stage high, its output will change state on each succeeding low to high transition of OUTPUT CLOCK. The second U101 stage in turn can only toggle on the positive going transition of OUTPUT CLOCK when its J and K inputs are high. Since the inputs are high at one-half the clock rate, by virtue of the first U101 stage, the second U101 stage toggles at one-fourth the OUTPUT CLOCK rate.

The two sections of U101, therefore, operate as a frequency divider, dividing the OUTPUT CLOCK by two when the write data is a "1" and by four when the data is a "0". Thus, in the low speed mode, four cycles of the 1200 Hz represent a "0" and eight cycles of 2400 Hz represent a "1". In the high speed mode, one cycle of 1200 Hz represents a "1" and one-half cycle of 600 represents a "0".

The output on pin 14 of U101 is applied to one section in U109 which provides sufficient current drive for the divider network. This divider and a jumper arrangement allow selecting one of three outputs to be fed to the audio output jack J6. The I-to-J jumper selects a 500 mv signal for the auxiliary input to an audio recorder; the I-to-H jumper selects a 50 mv signal for the microphone input to an audio recorder.

When the CDI is in the read mode, data from the recorders enters on J7. This input is fed to the negative input (pin 6) of operational amplifier U108.

The first section of U108 is a high gain amplifier, with its gain (approximately 100) being determined by R142 and R143. The output from this amplifier is coupled to input pin 2 of the following U108 stage and the base of a Darlington pair (Q4 and Q5) which provides high current gain.

Current into the base of transistor Q5 causes C67 to discharge. (C67 charges through R39 to 5 V dc.) The voltage on C67 in turn controls the gate of field effect transistor (FET) Q3. Q3 functions as a variable resistor which can be changed by its gate voltage. Since Q3 is connected between ground and the input network to the



first U108 stage, it serves as a variable shunt. A low gate voltage on Q3 decreases the shunt resistance and the input to U108. In a like manner, a high voltage on C67 results in an increased input to U108. Q3, Q4 and Q5 with their associated circuitry, therefore, serve as an automatic gain control (AGC) circuit which limits the input to the second U108 stage to approximately a positive 2 volt peak signal.

The second stage of U108 is a comparator with hysteresis that performs the needed audio to digital conversion. Feedback resistor R147, in conjunction with R145, establishes the level on the positive input (pin 3) of U108. This level, be it positive or negative, is the threshold voltage,  $\pm 50$  mv, which the negative input (pin 2) must exceed in order for the output of U108 to switch levels, positive to negative and the converse. Since the feedback loop is regenerative, U108 switches at its maximum rate, and U108 switches on each transition of the audio signal input. It is in this manner that U108 performs the audio to digital conversion.

The digital output of U108 is inverted in one section of inverter U109 and applied to pin 9 of exclusive OR gate U99 which is connected as a buffer without inversion. If the output of U109 is low, the output on pin 10 of U99 is also low and the output on pin 4 of another U99 exclusive OR gate is high. The voltage across C49 under this condition is minimal. When the output of U109 goes high, C49 starts to charge through R118 until pin 9 of U99 crosses the threshold of that gate. At this point pin 10 of U99 goes high, and since the two inputs to the second exclusive-OR gate are both high, pin 4 of U99 goes low. C49 now discharges because pins 9 and 10 of U99 are at the same level so that the circuit can repeat the operation on the next high to low transition at pin 4 of U109. R118, C49 and U99 consequently serve as a transition detector that produces a pulse less than one microsecond long for each transition of the output on pin 4 of U109, regardless of the polarity of the transition.

Transition pulses from U99 clock both D flip-flops in U113. A transition pulse clocks the top U113 at pin 3 which sets Q (pin 1) high and  $\bar{Q}$  (pin 2) low to enable up binary counter U112 on pin 15. Pin 1 is applied to the D input (pin 9) of the lower U113 and the circuit remains in this state until one of two things occurs: 1) a second transition pulse arrives before U112 reaches count 12 or 2) U112 reaches count 12.

If a second transition pulse arrives before count 12, the bottom U113 stage is set and presents a "1" to the D input (pin 9) of flip-flop U100. This is clocked by the  $\bar{Q}$  output on pin 2 of U113 as a low to pin 12 of U100.

If a transition pulse does not arrive before count 12, the bottom U113 stage outputs a "0" to input pin 9 of U100. On count 12, the C and D outputs of U112 go high to reset U113 by way of U98 at pin 4. As a result the U100 clock goes high, as does pin 12 of

U100. The output on pin 12 of U100 is inverted by U109 and applied to the receive input (pin 20) of the UART.

The Q output on pin 1 of U113, which occurs at the actual bit rate of the incoming data, is also used by the receive clock circuitry to reconstruct the receive clock from the data signal.

Received data undergoes serial-to-parallel conversion in the UART and is placed on the R01-8 data outputs of the UART when ROD (pin 4 of the UART) is low (PORT IN FB active) and onto INT0-7.

Four status outputs from the UART can also be enabled when SFD (pin 16) is low. These four bits are FE (framing error), OE (overrun error), DR (data ready) and TBRE (transmitter buffer register empy).

## 8.6 KEYBOARD

### 8.6.1 Block Diagram Analysis

A simplified block diagram of the keyboard is provided on Page X-25 in Section X.

The Clock Oscillator produces the basic timing signals for the keyboard, and they are distributed as indicated.

At the heart of the keyboard is a Key Switch Capacitive Matrix which can be viewed as a 16 x 16 X-Y matrix, with X being the column and Y the row. Conceptually, a key depression increases the capacitance between the X and Y coordinates that uniquely define that key.

The Column Scanner supplies a pulse train to the X lines in the matrix, with only one line being pulsed at any given point in time. When a key is depressed to increase the capacitance between the Column Scanner output and a Row Scanner input, the X-Y coordinates for that key are detected to provide an input to the Sense Circuit.

The Sense Circuit in turn generates an input to the Sequence Detector when a key closure occurs. This detector basically detects key closures and count cycles of the Row Scanner to discriminate against false key signals and insure that valid closures are serviced in order.

In the absence of key closures, the Sequence Detector feeds PKD to the Sense Circuit to increase its threshold. This action serves to increase the circuit's noise immunity. On valid key closures, the PKD input is such as to decrease the Sense Circuit's threshold. When valid key closures exist, the Sequence Detector strobes data into the Output Latch. The low order four bits to this latch are supplied by the Row Scanner; the high order four bits are

supplied by the Encoding ROM, with the data being determined by inputs from the Column Scanner and Function Latch Decoder. This strobe (Data Out) also enables the Strobe Generator to output STROBE, a 6 usec pulse that signals the Sol CPU that the Keyboard is ready to send data.

Eight bits of keyboard data (KBD0 through KBD7) are stored in the Output Latch. KBD0 through KBD6 represent the ASCII code for the character associated with the key closure, or closures, that initiated the Data Out strobe from the Sequence Detector. KBD7 is used only for special control characters (e.g. MODE SELECT, CLEAR and cursor movement) that are recognized by the Sol program. The data on KBD0-7 is input to the Sol CPU when it issues PORT IN FC (refer to Paragraph 8.5.2 on Page VIII-14).

The Repeat Counter is enabled when the REPEAT key and a character key in the Key Switch Capacitive Matrix are pressed at the same time. When this occurs, Key Out (initiated by the character key closure) is active, and the Repeat Counter generates a periodic Repeat Strobe. This strobe disables the Sequence Detector and causes the Strobe Generator to output repetitive STROBE pulses. Column 30 also prevents the Sequence Detector from strobing additional data into the Output Latch.

The Function Latch and Decoder latches and decodes the Low Order Count from the Row Scanner when the "function key" column in the Switch Matrix is selected by the Column Scanner. It then outputs, as appropriate, LOCAL, RST and BRK to J1 and SHIFT/SHIFT LOCK, UPPER CASE and CONTROL bits to the Encoding ROM. The latter three supply three of the seven address bits to the ROM which specify the high order four KBD bits (KBD4-7).

All keyboard outputs are supplied to J1 which is connected to J3 on the Sol-PC.

#### 8.6.2 Circuit Description

Refer to the Keyboard schematic in Section X, Page X-23.

Keyboard operation is controlled by a 3 usec clock circuit consisting of NAND gate U7, R7 and C7. U7 is connected as a Schmitt trigger inverter with negative feedback through R7 and C7. The output on pin 11 of U7, a square wave with a 3 usec period, is inverted in U4 (a NAND gate connected as a simple inverter) and applied to the clock input (pin 11) of U8. U8 operates in a toggle mode by virtue of feeding its  $\bar{Q}$  output on pin 8 to the D input on pin 12. Thus, its output state changes on each clock to produce a 6 usec and an inverted 6 usec clock on pins 9 and 8 respectively.

Each of these outputs is connected to a section of U7 where each is AND'ed with the 3 usec clock. This generates two negative going clocks at pins 8 and 6 of U7. These outputs are called  $\bar{\phi}1$  and

$\overline{\phi 2}$  respectively. This circuit thus generates a symmetrical two phase clock, with each phase having a 6 usec period with a 1.5 usec negative going pulse.

$\overline{\phi 1}$  advances the cascaded ripple counter, U5 and 6, in the Column Scanner circuit (U5, U6, NAND gates U4 and decoders U17 and U21). U6 divides  $\overline{\phi 1}$  by two on each advance. The output on pin 12 is consequently a square wave with a 12 usec period, the output on pin 9 is a square wave with a 24 usec period, and so on to pin 11 which has a 96 usec period. The output on pin 11 is then divided by two in U5 to provide 192, 384, 768 and 1536 usec periods. We will call these Clock 1 for the 12 usec period, Clock 2 for the 24 usec period, Clock 4 for the 48 usec period, and so on from Clock 8, 16, 32, 64 and 128.

Clocks 16, 32 and 64 are applied to the A, B and C inputs of binary-to-decimal decoders U17 and U21. In order for these decoders to yield outputs, their D inputs (pin 12) must be low. U4 is used to enable one or the other of these inputs, with Clock 128 being the determining factor. When Clock 128 is low, U17 is selected through U4 when  $\overline{\phi 1}$  is high at pin 4 of U4. U21 is selected when Clock 128 is high and  $\overline{\phi 1}$  is high at pin 13 of U4. By AND'ing  $\overline{\phi 1}$  and Clock 128, neither decoder is selected when  $\overline{\phi 1}$  is low, the time U5 and U6 count. During this time false binary signals can appear on the outputs of U5 and 6.

The net effect is that only one of the 15 outputs from U17 and 21 will be low, and this low advances on each count advance. The low outputs of U17 and 21 drive the column lines in the key switch matrix.

Clocks 1 through 8 are connected to analog multiplexers U19 and U22. Only one channel from input to output is connected at one time. Note that Clock 8 and Clock 8 from U6 enable U19 and U22 respectively. U19 and U22 (the Row Scanner) thus scan through the 16 rows in the sequence indicated by the numbers contained within the "boxes" of the key switch matrix. An entire scan of the rows is made before the next column is selected by U17 and 21.

We now have U17 and U21 driving the column lines and U19 and U22 testing each row line by connecting it to an input to the Capacitance Keyswitch (KTC) Detector. These two inputs are normally high at 5 volts. Within the switch matrix there is a small capacitance connected between each column and row line; that is, there is a capacitance associated with each key on the keyboard. When a key is depressed on the keyboard, the capacitance associated with that key increases. When the column and row lines associated with that key are selected, there is a significant voltage difference between the two and the capacitance charges to produce a small negative going spike at the input to the Capacitance Keyswitch Detector.

This detector circuit consists of three transistors, Q7, Q8, and Q9 (connected as a linear amplifier with negative feedback) followed by Q4 and Q2. Q4 and Q2 are large signal amplifiers biased in their cut-off region. The input to the detector is selectively connected to +5 V dc by way of the analog multiplexers (U19 and U22), the row matrix wires, and the 33K resistors. A key depression causes a negative current pulse through R16 to the base of the input amplifier transistor, Q8, which is biased near cut-off. The pulse is then amplified by Q8 with inversion to appear as a positive pulse at the input of Q7. Q7 is an emitter follower circuit which gives a positive pulse at its output, across R18, at a low impedance. This signal is coupled back to the input through transistor Q9, a common base amplifier which has its base clamped to 2.5 V dc by zener diode CR4. When the positive pulse appears at the emitter of Q9, it is amplified without inversion and applied to the input of Q8. Since the original input was a negative pulse, the positive pulse constitutes negative feedback. The output across R18, a positive pulse, is further amplified by pulse amplifier transistor Q4, a common base amplifier that is normally biased off. The output stage Q2 is biased in the cut-off region also, but a sufficient positive pulse from Q4 will cause Q2 to conduct to give a negative pulse output across R12.

Transistors Q1, Q6, Q5 and Q3, represent a second pulse amplifier circuit that is analogous to transistors Q9, Q8, Q7 and Q4 respectively. The output of this second amplifier, which appears at the collector of Q3, is also connected to the base of the output transistor Q2. An input pulse from either U19 or U22 will therefore supply an amplified negative pulse to pin 13 of NOR gate U14.

The  $\overline{\text{PKD}}$  signal through R24 helps to set the threshold at the base of Q4 and Q3. This threshold is normally high when  $\overline{\text{PKD}}$  is high, so the output from Q7 and Q5 has to overcome a higher threshold at the emitter of Q4 and Q3 in order to cause conduction of Q4 and Q3. On the second such pulse on the same count address,  $\overline{\text{PKD}}$  goes low to reduce the threshold at the bases of Q4 and Q3. This sensitizes the circuit, acting as a positive feedback path, and gives an output. Thus two consecutive detections of a key stroke are necessary to give an output. This feature provides noise immunity since a single noise pulse will not pass through the amplifier. The complete key switch matrix is scanned at a very high rate compared to the time it takes to physically press and release a key. Thus a key closure will be detected, even though the key is not held down for any appreciable time.

Two sections of NOR gate U14 are connected as a cross-coupled flip-flop. A low on pin 13 of U14 sets output pin 11 of U14 high, providing that the low is longer than 1.5 usec (which it is when a valid key closure is detected). That is because  $\overline{\text{QI}}$  is applied to pin 9 of U14.  $\overline{\text{QI}}$  effectively prevents switching noise, which is short in duration, from being interpreted as a key closure. The high, let's call it KEY, on pin 11 of U14 will remain until  $\overline{\text{QI}}$  again goes low about 4.5 usec later.

KEY is fed to pin 5 of 8-input NAND gate U25, pin 9 of ROM U20 and pin 1 of NAND gate U27. Let's examine the other inputs to U25.

KEY, as mentioned, is fed to pin 9 of U20 which is a 256 x 4 bit static ROM. Only two bits are used. For each possible row-column combination, there is one storage location in U20. DI1 and DO1 (pins 9 and 11) are the input and output respectively of one bit location; DI2 and DO2 serve the same functions for the other bit location. The row count is applied to A0-4 and the column count is applied to A5-7 to address U20.

When a key closure is detected, the counts are presented to U20 continuously. When the counts change shortly after the falling edge of  $\phi 1$ , U20 outputs the status of the address that is already stored in the ROM about 1 usec later on pin 10. On the rising edge of  $\phi 1$  after the address change, the status on pin 10 is latched in one-half of D flip-flop U26 and presented at output pins 9 and 8. About 1.5 usec later the R/W signal on pin 20 of U20 goes low, and the KEY signal on pin 9 enters the specified location in U20. Note that this KEY is related with the new count address. The key stored in U26 represents the preceding address. We consequently call the KEY in U26 "KEY minus 1", and it is applied to pin 11 of U25.

The remaining inputs to U25 are 1)  $\phi 2$  (an inverted  $\phi 2$ ) on pin 12, 2) a repeat strobe signal on pin 4 (supplied by pin 11 of NAND gate U16 which is high without a repeat command), 3) PKD minus 1 on pin 6 (supplied on pin 3 of U26 which is low if three or more count cycles have occurred since one key closure), and 4) the column output on pin 4 of U17 which is applied to pins 1, 2 and 3. The last signal drives the column associated with the special function keys on the keyboard (SHIFT, SHIFT LOCK, LOCAL, BREAK, UPPER CASE, REPEAT and CONTROL).

In order for U25 to output a low on pin 8, therefore, we need a current KEY, a KEY from the preceding count cycle, no repeat function, no drive on pin 4 (column 30, hexadecimal), and we must be on the second count cycle during the current key depression.

With these conditions satisfied output pin 8 of U25 goes low. It is inverted by U10 to a high on pin 11. This signal then clocks the output latches, U1 and 2. On this signal, the data present on the inputs are latched into U1 and 2, and it remains latched until the next output on pin 8 of U25 occurs.

A low on pin 8 of U25 also resets one-half of D flip-flop U11 at pin 13 which causes output pin 9 to go low. On the rising edge of the inverted 6 usec clock from U8, the second U11 stage sets and out-



put pin 5 goes low to clear the first stage. The high on output pin 6 is inverted by NAND gate U10 to supply a low active STROBE on pin 3 of J1. (Note that J1 on the keyboard connects to J3 on the Sol-PC.) The next inverted 6 usec clock resets the second U11 stage. We thus have a 6 usec strobe pulse following the latching of data into U1 and U2.

The complement of KEY minus 1 on output pin 8 of U26 is fed to input pin 10 of NAND gate U16 and is translated to a high on pin 8. The other input on pin 9 is high at this time since it is driven by the signal which indicates the third count cycle. A three-input NAND gate, U27, thus has a high on pin 2. A second input on pin 1 is KEY which is active (high) from the first count cycle of the key closure. The remaining input on pin 13 is supplied by pin 11 of U16, and it is low only when the repeat function is operating. U27 is consequently satisfied and outputs a low on pin 12.

This low appears at pin 5 of NOR gate U16. Pin 4 of U5 is high at this point by virtue of a low on pin 1 of U16 which indicates the third count. Thus, the high on pin 6 of U16 will be stored in the second bit location U20 when  $\overline{\phi}2$  goes low at pin 20 of U20. When this happens DO2 (pin 12) of U20 goes high to indicate the new status of this bit.

The DO2 output is inverted in U10 and applied to input pin 2 of another U26 D flip-flop and to the Capacitance Keyswitch Detector as PKD. PKD serves to lower the detector threshold; that is, the detector offers less "resistance" to its input. This is positive feedback that allows the detector to discriminate between noise and a key closure. Note that two key closures are required before the detector threshold is lowered.

The inverted DO2 output from U20 also appears at the D input (pin 2) of U26. Since this flip flop is clocked by  $\overline{\phi}1$ , the prior status of PKD, called "PKD minus 1", is already present in this latch on output pin 5. If we are on the second count cycle of a key closure, pin 5 is high. If we are on the third count or more, it is low to inhibit U25. As previously mentioned, PKD minus 1 is also connected to the NOR gate (U16) used to feed data to pin 11 of U20 from KEY minus 1.

When the current KEY signal is released, pin 12 of NAND gate U27 and pin 5 of NAND gate U16 go high. The U16 NAND gate that inputs to pin 4 of U16 looks at KEY minus 1 on pin 2 and the complement of PKD minus 1 on pin 1. Thus, pin 1 is high for the first one and a half counts and pin 2 is high for the first count. Upon release of KEY, therefore, pin 3 of U16 is low for the first count. On the second count, KEY minus 1 goes low--as do pin 6 of U16 and pin 12 of U20. On the next  $\overline{\phi}2$  clock, the data is read into U20. The output on pin 12 of U20 changes to remove PKD which increases the Capacitance Keyswitch Detector threshold for greater noise immunity. It also sets PKD minus 1 on pin 5 of U26 on the third count cycle

following release of KEY. On the third cycle the circuit reverts to its original state.

This circuit, comprised of U20, U26, U16 and U17 serves two functions. By requiring two events during two consecutive count cycles before generating a KEY, it discriminates against false key closures. It also insures that multiple key strokes are serviced in order. (This is the n-key rollover feature.) That is because the row-column addresses are continuously presented to U20 and this circuit's cycle can occur for each possible key closure. U20 can thus contain data for all possible key closures, and the data will enter U1 and U2 on the KEY generated for each closure as the row-column count progresses.

The previously mentioned column 30 output on pin 4 of U17 drives the keyboard control key "switches". Data for these key closures, present on pins 1, 2 and 3 of addressable latch U12 is latched in U12 during Clock 8 and 02 when column 30 is driven. Pin 13 of U12 is connected to the complement of PKD minus 1. Thus, the data (active low) is strobed into U12 on the first count cycle. During the third count it will be strobed again and a high is read in. When the key is released, a low is strobed in again. As a result, a high active pulse appears on the output line related to the key that was closed for the duration of the key closure.

SHIFT and SHIFT LOCK, on pins 11 and 10 respectively, are applied through U23 inverter stages to NOR gates U13 and U14. These are connected as a cross-coupled flip-flop. An active SHIFT sets this flip-flop at pin 5 of U13 to make output pin 6 of U13 and output pin 3 of U14 high. The latter is connected to pin 3 of U18, a 512 x 4 bit ROM. U18 is programmed to output the high-order four bits of the data to U1 according to the states of pins 1, 2 or 3.

The U13-14 flip-flop is set to a high on pin 6 if SHIFT LOCK is active. As can be seen, the shift bit to U18 is high by virtue of the low on pin 6 of U13 and it will remain so until SHIFT again causes U13-14 to change state. When output pin 6 of U14 is high, pin 12 of U24 is low to turn light emitting diode LED1 on. This LED is located in the SHIFT LOCK key and indicates the keyboard is in a locked shift condition.

When UPPER CASE is active, pin 7 of U12 goes high to clock D flip-flop U15 on pin 3. This flip-flop is connected to operate in a toggle mode. On the UPPER CASE "clock", pin 5 of U15 goes to make pin 2 of U18 low. The high on pin 6 of U15 is inverted by U24 to turn on LED2. LED2 is located in the UPPER CASE key. A second closure of this key toggles U15 to the opposite condition.

Now assume the LOCAL key is depressed, the output on pin 5 of U12 goes active high to clock the other D flip-flop U15 stage at pin 11. This stage also operates as a toggle, and output pin 9 goes low to become LOCAL on pin 14 of J1. Again, the high on output pin 8

causes LED3, the LOCAL light, to turn on. A second closure of the LOCAL key toggles this section of U15 to the opposite condition. Note that LOCAL has no affect on keyboard data.

The other outputs from U12 are BREAK (pin 12), CONTROL (pin 6) and REPEAT (pin 9). BREAK is inverted in U23 to become BRK on pin 4 of J1. CONTROL is applied directly to input pin 1 of U18 so that the control character related to the low order bits enters U1 and U2.

REPEAT is applied to pins 10 and 11 of NAND gate U27 and pin 13 of NAND gate U16. The input to U27 is gated with UPPER CASE to generate RST at pin 13 of J1. This means, of course, that REPEAT and UPPER CASE must be depressed at the same time to generate RST.

On pin 13 of U16, REPEAT enables that gate so that U16 transmits the output on pin 9 of U9. U9 is connected as a two-stage shift register whose input (pin 2) is ground. It is clocked by clock 128 from U5.

U9 is initially set with output pins 5 and 9 high during the third count cycle by PKD minus 1. This is also the time when U12 outputs data. If the key is released, U9 clears to a low on pin 9 five count cycles following KEY. If the key is held down, U9 cannot shift since PKD minus 1 remains on preset input pins 4 and 10.

When REPEAT exists at pin 13 of U16, pin 11 of U16 is low to inhibit U25 and U27 at pin 13. This prevents further KEY signals and disables the n-key rollover circuitry. The low on pin 11 of U16 is also inverted by open collector inverter U24 to enable the repeat oscillator (timer U3, R4, R5 and C3). U3 generates a square wave on pin 3 with a period determined by the RC network.

This clocks the first stage of D flip-flop U11, the STROBE generator, and U11 produces the previously discussed 6 usec STROBE. U11 continues to generate STROBE at the repeat oscillator rate until either the REPEAT or character key is released. And with each STROBE, of course, the data associated with the character key is latched into U1 and U2.

Eight ASCII-coded data bits are output by U1 and U2 to J1 as indicated. Seven bits (0-6) are used for ASCII characters, and the eighth bit (7) is set only for certain control characters that are recognized by the Sol program. These are used for control functions such as MODE SELECT and cursor movement.

The remaining circuit, R32 and C14, initializes the keyboard when power is applied. That is, it resets the output latches and the SHIFT/SHIFT LOCK, UPPER CASE and LOCAL flip-flops. It also inhibits STROBE at pin 1 of NAND gate U10.



## IX SOFTWARE

9.1	CONSOL . . . . .	IX-1
9.1.1	DUMp (addr) (addr) . . . . .	IX-2
9.1.2	ENter addr . . . . .	IX-2
9.1.3	TLoad (speed) . . . . .	IX-2
9.1.4	EXecute addr . . . . .	IX-3
9.1.5	BAsic . . . . .	IX-3
9.2	Standard I/O Routines . . . . .	IX-3

SOLOS<sup>TM</sup>/CUTER<sup>TM</sup> USERS' MANUAL  
(Seperately bound. May be inserted here.)

BASIC 5 USERS' MANUAL  
(Seperately bound. May be inserted here.)

SOLOS<sup>TM</sup> Monitor Program Source Listing

CONSOL<sup>TM</sup> Monitor Program Source Listing

## 9.1 CONSOL

If you have SOLOS refer to the SOLOS/CUTERS USERS MANUAL, and ignore Section 9.1 and 9.2. CONSOL is a 1024 byte program designed to allow the Sol TERMINAL/COMPUTER to operate as a standard CRT terminal and to provide access to the essential computer capabilities of the Sol. This in addition to providing verification of correct system operation helps in finding errors in case of a malfunction.

In addition, CONSOL contains standardized entry points for all normal I/O operations. These routines are common with all Sol System Software allowing each personality module in the Sol line to interface with external programs in an almost identical manner.

A cassette read routine is also resident in the CONSOL module allowing Sol Software to be loaded and run in a system with additional memory. Sol System Software as of November 1976 includes BASIC, FOCAL, a Scientific Calculator and numerous "game" packages including a 8K assembly language version of STARTREK called TREK80.

When power is applied to the Sol unit, CONSOL initializes the system RAM area, clears the screen, and enters the terminal mode.

In this mode the Sol System acts as a standard CRT terminal sending keyboard data to an output port and displaying received data on the screen. The COMMAND KEYS of the keyboard are not transmitted to the output port but are interpreted as direct internal operation keys. CURSOR MOVEMENT, HOME and CLEAR SCREEN all operate in this manner, while MODE SELECT causes an immediate change in the operation of the unit.

When the MODE key is depressed CONSOL issues a prompt character (>) and waits for a command line to be input. The Sol is now operating as a computer and is ready to accept one of the following commands:

DUMp	Dump memory locations to screen
ENTER	Enter data to memory
EXECute	Execute a program in external memory
BASic	Execute a program located at address zero
TERminal	Return to terminal mode
TLoad	Load program or data from cassette tape
MODE	Press MODE SELECT key to start new command line

Try using the commands as described in the following pages.

## 9.1.1 DUMP (addr) (addr)

The DUMP command displays memory data on the screen in a Hexidecimal representation. As with all Sol commands the command is recognized by the first two characters and up to ten additional characters can be input without an error being forced.

Thus, DU; DUST; DUMP; DUMPTHESE would all be recognized as being a DUMP command.

At least one address must follow the command or a error displayed on the screen. If two addresses are input then all values from the first address to the last will be displayed.

DUMP Ø EF

Up to ten blanks may be inserted between each parameter without forcing an error condition. Errors are indicated by a question mark (?) replacing the character where the error occurred. For example if the DU command were given without an address the question mark would appear ten spaces to the right of the "U".

## 9.1.2 ENTER addr

The ENTER command places sequential bytes into memory beginning at the specified address. Data, represented as hexadecimal values, are input from the keyboard for entry to memory. All CONSOL commands except MODE SELECT are executed when the RETURN key is pressed. After the ENTER, (address), RETURN sequence the Sol Displays a colon (:) prompt character. Values are then input one line at a time with each line terminated by a carriage return or linefeed. The ENTER function itself is terminated with a slash (/) and the Sol goes back to the command mode when the slash is encountered.

With all command functions of CONSOL, input lines are terminated with a carriage return or line feed. If the terminator is a C/R, CONSOL will erase all characters from the current cursor location to the end of the screen line. In this case, all valid input should be to the left of the cursor. If an error occurred during input the cursor may be moved to the left using the "cursor-left" key and the erroneous characters changed. A linefeed would then be used as a terminator since LF does not erase the line prior to processing the characters. This is particularly useful when using the ENTER command since the input line can be visually scanned and errors corrected prior to the actual entry of input data to memory.

## 9.1.3 TLoad (speed)

Included within CONSOL are routines to read standardized cassette tape Software which is recorded with a sixteen byte header that includes NAME, LOAD INFORMATION, FILE TYPE and execute address. CONSOL, because of space limitations, is unable to search for a



program or file by name. After receiving the TLoad command, CONSOL turns on the cassette player and waits for the next header, then uses the header information and loads the file to memory. The cassette recorder must be in play mode and properly connected before executing the TLoad command.

After loading the data, CONSOL returns to the command mode where the EXEC command can be used to execute the just loaded program. Also, a return can normally be made to the command mode by pressing the MODE SELECT key. Space limitations again limited escape during the header search, so if the system locks up in this routine the standard Sol restart must be used. To restart the Sol press UPPER CASE and REPEAT keys simultaneously.

The CUTS cassette interface electronics within the Sol will record or receive data at either of two standard speeds. TLoad will accept a parameter to select this speed, 0 being high speed and 1 being low. (1200 and 300 bits per second respectively). If no parameter is given CONSOL will default to high speed operation as all standard Processor Technology Sol-System Software is recorded at this speed.

#### 9.1.4 EXecute addr

The execute command is used to run programs located in external memory. CONSOL branches to the external routine in a manner similar to an 8080 CALL instruction so the program can return to the command mode using a standard 8080 RET instruction if normal stack operations are used.

#### 9.1.5 BASic

The BASic command is provided for executing programs whose starting address is 0, such as Sol-BASIC5.

### 9.2 STANDARD I/O ROUTINES

All Sol System personality modules contain similar I/O code for input/output operations. CONSOL, using 1K of memory, has routines for KEYBOARD and SERIAL PORT input as well as Serial Communications Channel and VIDEO DISPLAY OUTPUT. Although the same code for SOLOS and SOLED contains expanded functions, the I/O operations appear almost identical when used with external software.

Sol-BASIC5, for example, performs all I/O using the jump table of the personality modules. Thus, without altering BASIC the user may output to either the serial port or to the display screen. Provision is also made within BASIC to programatically change to any of the four available Input or Output options. CONSOL is of course limited to the two provided.

SOLOS<sup>TM</sup> Monitor Program Source Listing

```
C000 00
C001 C3 AF C1
```

```
C026 E6 03
C028 07
C029 85
C02A 6F
C02B C3 27 C2
```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

```

0105 * DATA AVAILABLE. IF NO CHARACTER HAS BEEN RECEIVED THE
0106 * ROUTINE RETURNS WITH THE ZERO FLAG SET. OTHERWISE THE
0107 * CHARACTER IS INPUT AND A RETURN MADE WITH THE CHARACTER
0108 * IN THE ACCUMULATOR AND THE ZERO FLAG RESET.
0109 *
0110 *
0111 *      KEYBOARD INPUT DRIVER
0112 *
0113 KSTAT IN STAPT GET STATUS WORD
0114 CMA . INVERT IT FOR PROPER RETURN
0115 ANI KDR TEST KEYBOARD BIT
0116 RZ . ZERO IF NO CHARACTER RECEIVED
0117 *
0118 IN KDATA GET CHARACTER
0119 RET . GO BACK WITH IT
0120 *
0121 *
0122 * THIS JUMP IS PART OF THE AUTO START UP CODE
0123 *
0124 DB 0 VERIFY ADDR=C037 SO NEXT INSTRUCTION IS AT C038
0125 JMP INIT
0126 *
0127 *
0128 *      JUMP TABLE OUTPUT ROUTINES
0129 *
0130 * THIS ROUTINE SETS UP THE DISPATCH TABLE FOR OUTPUT
0131 * ROUTINES. THE CHARACTER FOR OUTPUT IS IN REGISTER "B".
0132 * OUTPUT IS MADE TO THE DRIVER POINTED TO BY THE REGISTER
0133 * "A". THE DEVICE DRIVERS ARE DEFINED AS FOLLOWS:
0134 *
0135 *
0136 * 0 - DISPLAY SCREEN
0137 * 1 - SERIAL OUTPUT PORT
0138 * 2 - PARALLEL OUTPUT PORT
0139 * 3 - USER DEFINED OR ERROR FLAG
0140 *
0141 * ENTRY AT: SOUT SELECTS CURRENT OUTPUT DEVICE
0142 * AOUT SELECTS DEVICE IN REGISTER "A"
0143 *
0144 OUTPR PUSH H
0145 LXI H,OTAB POINT TO OUTPUT TABLE
0146 JMP IOPRC AND DISPATCH TO OUTPUT ROUTINE
0147 *
0148 *
0149 *
0150 *
0151 *      SERIAL INPUT DRIVER
0152 *
0153 SSTAT IN SERST GET SERIAL STATUS WORD
0154 ANI SDR TEST FOR SERIAL DATA READY
0155 RZ . FLAGS ARE SET
0156 *
0157 IN SDATA GET DATA BYTE

```

C03B E5  
C03C 21 92 C2  
C03F C3 26 C0

C042 DB F8  
C044 E6 40  
C046 C8

C047 DB F9

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 2

```

C049 C9
0158 RET WE HAVE IT
0159 *
0160 *
0161 *      SERIAL DATA OUTPUT
0162 *
0163 SDROT IN SERST GET PORT STATUS
0164 RAL . PUT HIGH BIT IN CARRY
0165 JNC SDROT LOOP UNTIL TRANSMITTER BUFFER IS EMPTY
0166 MOV A,B GET THE CHARACTER BACK
0167 OUT SDATA SEND IT OUT
0168 RET AND WE'RE DONE
0169 *
0170 *
0171 *
0172 *
0173 *
0174 *      VIDEO DISPLAY ROUTINES
0175 *
0176 *
0177 * THESE ROUTINES ALLOW FOR STANDARD VIDEO TERMINAL
0178 * OPERATIONS. ON ENTRY, THE CHARACTER FOR OUTPUT IS IN
0179 * REGISTER B AND ALL REGISTERS EXCEPT "A" AND FLAGS ARE
0180 * UNALTERED ON RETURN.
0181 *
0182 *
0183 *
0184 VDMOT PUSH H SAVE MOST REGISTERS
0185 PUSH D
0186 PUSH B
0187 *
0188 *      TEST IF ESC SEQUENCE HAS BEEN STARTED
0189 *
0190 LDA ESCFL GET ESCAPE FLAG
0191 ORA A
0192 JNZ ESCS IF NON ZERO GO PROCESS THE REST OF THE SEQUENCE
0193 *
0194 *
0195 CHPCX MOV A,B SAVE IN B...STRIP PARITY BEFORE SCREEN!
0196 ANI 7FH CLR PARITY TO LOCATE IN TBL
0197 MOV B,A KEEP IT W/OUT PARITY IN R TOO
0198 JZ GOBK DO A QUICK EXIT IF A NULL
0199 LXI H,TBL POINT TO SPECIAL CHARACTER TABLE
0200 CALL TSRCH GO PROCESS
0201 *
0202 GOBACK CALL VDADD GET SCREEN ADDRESS
0203 MOV A,M GET PRESENT CURSOR CHARACTER
0204 ORI 80H
0205 MOV M,A CURSOR IS BACK ON
0206 LHLD SPEED-1 GET DELAY SPEED
0207 INR L MAKE SURE IT IS NON-ZERO
0208 XRA A DELAY WILL END WHEN H=0
0209 TIMER DCX H TIMER DELAYS HERE
0210 CMP H DONE WITH DELAY YET

```

C04A DB F8  
C04C 17  
C04D D2 4A C0  
C050 78  
C051 D3 F9  
C053 C9

C054 E5  
C055 D5  
C056 C5

C057 3A 0C C8  
C05A B7  
C05B C2 5F C1

C05E 78  
C05F E6 7F  
C061 47  
C062 CA 7C C0  
C065 21 73 C2  
C068 CD 82 C0

C06B CD 1C C1  
C06E 7E  
C06F F6 80  
C071 77  
C072 2A 0A C8  
C075 2C  
C076 AF  
C077 2B  
C078 BC

PROGRAM DEVELOPMENT SYSTEM

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

```

C079 C2 77 C0      0211      JNZ      TIMER  KEEP DELAYING
C07C C1             0212 GOBK      POP      B
C07D D1             0213      POP      D      RESTORE REGISTERS
C07E E1             0214      POP      H
C07F C9             0215      RET      .      EXIT FROM VDMOT
                  0216 *
C080 23             0217 NEXT     INX      H
C081 23             0218      INX      H
                  0219 *
                  0220 *
                  0221 * THIS ROUTINE SEARCHES THROUGH A SINGLE CHARACTER
                  0222 * TABLE FOR A MATCH TO THE CHARACTER IN "B". IF FOUND
                  0223 * A DISPATCH IS MADE TO THE ADDRESS FOLLOWING THE MATCHED
                  0224 * CHARACTER. IF NOT FOUND THE CHARACTER IS DISPLAYED ON
                  0225 * THE MONITOR.
                  0226 *
C082 7E             0227 TSRCH    MOV      A,M    GET CHR FROM TABLE.
C083 B7             0228      ORA      A
C084 CA 94 C0       0229      JZ      CHAR    ZERO IS THE LAST
C087 B8             0230      CMP      B      TEST THE CHR
C088 23             0231      INX      H      POINT FORWARD
C089 C2 80 C0       0232      JNZ     NEXT
C08C E5             0233      PUSH    H      FOUND ONE...SAVE ADDRESS
C08D CD 36 C1       0234      CALL    CREM    REMOVE CURSOR
C090 E3             0235      XTHL    .      GET DISPATCH ADDRESS TO HL
C091 C3 27 C2       0236      JMP     DISPT  DISPATCH NOW
                  0237 *
                  0238 * PUT CHARACTER TO SCREEN
                  0239 *
C094 78             0240 CHAR     MOV      A,B    GET CHARACTER
C095 FE 7F          0241      CPI      7FH    IS IT A DEL?
C097 C8             0242      RZ      .      GO BACK IF SO
                  0243 *
                  0244 *
                  0245 *
C098                0246 OCHAR    EQU      $      ACTUALLY PUT CHAR TO SCREEN NOW
C098 CD 1C C1       0247      CALL    VDADD   GET SCREEN ADDRESS
C09B 70             0248      MOV      M,B    PUT CHR ON SCREEN
                  0249 *
C09C 3A 08 C8       0250      LDA      NCHAR   GET CHARACTER POSITION
C09F FE 3F          0251      CPI      63      END OF LINE?
COA1 DA C1 C0       0252      JC      OK
COA4 3A 09 C8       0253      LDA      LINE
COA7 FE 0F          0254      CPI      15      END OF SCREEN?
COA9 C2 C1 C0       0255      JNZ     OK
                  0256 *
                  0257 * END OF SCREEN...ROLL UP ONE LINE
                  0258 *
COAC AF             0259 SCROLL    XRA      A
COAD 32 08 C8       0260      STA      NCHAR   BACK TO FIRST CHAR POSITION
COB0 4F             0261 SROL      MOV      C,A
COB1 CD 23 C1       0262      CALL    VDAD    CALCULATE LINE TO BE BLANKED
COB4 AF             0263      XRA      A

```

PROGRAM DEVELOPMENT SYSTEM

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 3

```

COB5 CD FA C0       0264      CALL    CLIN1   CLEAR IT
COB8 3A 0A C8       0265      LDA      BOT
COBB 3C             0266      INR      A
COBC E6 0F          0267      ANI      0FH
COBE C3 EE C0       0268      JMP     ERAS3
                  0269 *
                  0270 * INCREMENT LINE COUNTER IF NECESSARY
                  0271 *
COC1 3A 08 C8       0272 OK      LDA      NCHAR   GET CHR POSITION
COC4 3C             0273      INR      A
COC5 E6 3F          0274      ANI      3FH    MOD 64 AND WRAP
COC7 32 08 C8       0275      STA      NCHAR
COCA C0             0276      RNZ      .      DIDN'T HIT END OF LINE, OK
                  0277 PDOWN    EQU      $      CURSOR DOWN ONE LINE HERE
COCB 3A 09 C8       0278      LDA      LINE   GET THE LINE COUNT
COCE 3C             0279      INR      A
COCF E6 0F          0280 CURSC   ANI      0FH    MOD 15 INCREMENT
COD1 32 09 C8       0281 CUR      STA      LINE   STORE THE NEW
COD4 C9             0282      RET
                  0283 *
                  0284 * ERASE SCREEN
                  0285 *
COD5 21 00 CC       0286 PERSE    LXI      H,VDMMEM  POINT TO SCREEN
COD8 36 A0          0287      MVI      M,80H+  THIS IS THE CURSOR
                  0288 *
CODA 23             0289      INX      H      BUMP 1ST
CODB 36 20          0290 ERAS1    EQU      $      LOOPS HERE TO ERASE SCREEN
CDDD 23             0291      MVI      M,  .    BLANK IT OUT
CODE 7C             0292      INX      H      NEXT
CODF FE D0          0293      MOV      A,H    SEE IF END OF SCREEN YET
COE1 DA DB C0       0294      CPI      ODOH   ?
COE4 37             0295      JC      ERAS1   NO--KEEP BLANKING
                  0296      STC      .      CARRY WILL SAY COMPLETE ERASE
                  0297 *
COE5 3E 00          0298 PHOME    MVI      A,0      RESET CURSOR--CARRY=ERASE, ELSE HOME
COE7 32 09 C8       0299      STA      LINE   ZERO LINE
COEA 32 08 C8       0300      STA      NCHAR   LEFT SIDE OF SCREEN
COED D0             0301      RNC      .      IF NO CARRY, WE ARE DONE WITH HOME
                  0302 *
COEE D3 FE          0303 ERAS3    OUT      DSTAT  RESET SCROLL PARAMETERS
COF0 32 0A C8       0304      STA      BOT    BEGINNING OF TEXT OFFSET
COF3 C9             0305      RET
                  0306 *
                  0307 *
COF4 CD 1C C1       0308 CLINE    CALL    VDADD   GET CURRENT SCREEN ADDRESS
COF7 3A 08 C8       0309      LDA      NCHAR   CURRENT CURSOR POSITION
COFA FE 40          0310 CLIN1   CPI      64      NO MORE THAN 63
COFC D0             0311      RNC      .      ALL DONE
COFD 36 20          0312      MVI      M,  .    ALL SPACED OUT
COFF 23             0313      INX      H
C100 3C             0314      INR      A
C101 C3 FA C0       0315      JMP     CLIN1   LOOP TO END OF LINE
                  0316 *

```

PROGRAM DEVELOPMENT SYSTEM

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

```

0317 *
0318 * ROUTINE TO MOVE THE CURSOR UP ONE LINE
0319 *
0320 PUP LDA LINE GET LINE COUNT
0321 DCR A
0322 JMP CURSC MERGE TO HANDLE CURSOR
0323 *
0324 * MOVE CURSOR LEFT ONE POSITION
0325 *
0326 PLEFT LDA NCHAR
0327 DCR A
0328 PCUR EQU $ CURSOR ON SAME LINE
0329 ANI 3FH LET CURSOR WRAP
0330 STA NCHAR UPDATED CURSOR
0331 RET
0332 *
0333 * CURSOR RIGHT ONE POSITION
0334 *
0335 PRIT LDA NCHAR
0336 INR A
0337 JMP PCUR
0338 *
0339 * ROUTINE TO CALCULATE SCREEN ADDRESS
0340 *
0341 * ENTRY AT: RETURNS:
0342 *
0343 * VDADD CURRENT SCREEN ADDRESS
0344 * VDAD2 ADDRESS OF CURRENT LINE, CHAR 'C'
0345 * VDAD LINE 'A', CHARACTER POSITION 'C'
0346 *
0347 VDADD LDA NCHAR GET CHARACTER POSITION
0348 MOV C,A 'C' KEEPS IT
0349 VDAD2 LDA LINE LINE POSITION
0350 VDAD MOV L,A INTO 'L'
0351 LDA BOT GET TEXT OFFSET
0352 ADD L ADD IT TO THE LINE POSITION
0353 RRC . TIMES TWO
0354 RRC . MAKES FOUR
0355 MOV L,A L HAS IT
0356 ANI 3 MOD THREE FOR LATER
0357 ADI <VDMEM LOW SCREEN OFFSET
0358 MOV H,A NOW H IS DONE
0359 MOV A,L TWIST L'S ARM
0360 ANI OCOH
0361 ADD C
0362 MOV L,A
0363 RET H & L ARE NOW PERVERTED
0364 *
0365 * ROUTINE TO REMOVE CURSOR
0366 *
0367 CREM CALL VDADD GET CURRENT SCREEN ADDRESS
0368 MOV A,M
0369 ANI 7FH STRIP OFF THE CURSOR

```

PROGRAM DEVELOPMENT SYSTEM

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 4

```

0370 MOV M,A
0371 RET
0372 *
0373 * ROUTINE TO BACKSPACE
0374 *
0375 PBACK CALL PLEFT
0376 CALL VDADD GET SCREEN ADDRESS
0377 MVI M, PUT A BLANK THERE
0378 RET
0379 *
0380 * ROUTINE TO PROCESS A CARRIAGE RETURN
0381 *
0382 PCR CALL CLINE CLEAR FROM CURRENT CURSOR TO END OF LINE
0383 * REG A COMES BACK W/64 AND WILL BE CLEARED BY PCUR
0384 JMP PCUR AND STORE THE NEW VALUE
0385 *
0386 * ROUTINE TO PROCESS LINEFEED
0387 *
0388 PLF LDA LINE GET LINE COUNT
0389 INR A
0390 ANI 15 SEE IF IT WRAPPED AROUND
0391 JNZ CUR NO--NO NEED TO SCROLL
0392 JMP SROL YES--THEN SCROLL
0393 *
0394 * SET ESCAPE PROCESS FLAG
0395 *
0396 PESC MVI A,-1
0397 STA ESCFL SET FLAG
0398 RET
0399 *
0400 * PROCESS ESCAPE SEQUENCE
0401 *
0402 ESCS CALL CREM REMOVE CURSOR
0403 CALL ESCSP PROCESS THE NEXT PART OF SEQUENCE
0404 JMP GOBACK
0405 *
0406 ESCSP LDA ESCFL GET ESCAPE FLAG
0407 CPI -1 TEST FLAG
0408 JZ SECOND
0409 *
0410 * PROCESS THIRD CHR OF ESC SEQUENCE
0411 *
0412 LXI H,ESCFL
0413 MVI M,0 NO MORE PARTS TO THE SEQUENCE
0414 CPI 2
0415 JC SETX SET X IF IS ONE
0416 JZ SETY SET Y IF IS TWO
0417 CPI 8
0418 JZ STSPD SET NEW DISPLAY SPEED IF "8"
0419 CPI 9
0420 JC OCHAR PUT IT ON THE SCREEN
0421 RNZ
0422 *

```



\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

```

0423 * TAB ABSOLUTE TO VALUE IN REG B
0424 *
0425 SETX MOV A,B GET CHARACTER
0426 JMP PCUR
0427 *
0428 * SET CURSOR TO LINE "B"
0429 *
0430 SETY MOV A,B
0431 JMP CURSC
0432 *
0433 *
0434 * PROCESS SECOND CHR OF ESC SEQUENCE
0435 *
0436 SECOND MOV A,B GET WHICH
0437 CPI 3
0438 JZ CURET RETURN CURSOR PARAMETERS
0439 CPI 4
0440 JNZ ARET2
0441 *
0442 * ESC <4> RETURN ABSOLUTE SCREEN ADDRESS
0443 *
0444 ARET MOV B,H
0445 MOV C,L PRESENT SCREEN ADDRESS TO BC FOR RETURN
0446 *
0447 ARET1 POP H RETURN ADDRESS
0448 POP D OLD B
0449 PUSH B
0450 PUSH H
0451 XRA A
0452 ARET2 STA ESCFL
0453 RET
0454 *
0455 *
0456 * RETURN PRESENT SCREEN PARAMETERS IN BC
0457 *
0458 CURET LXI H,NCHAR
0459 MOV B,M CHARACTER POSITION
0460 INX H
0461 MOV C,M LINE POSITION
0462 JMP ARET1
0463 *
0464 *
0465 * START UP SYSTEM
0466 *
0467 * CLEAR SCREEN AND THE FIRST 256 BYTES OF GLOBAL RAM
0468 * THEN ENTER THE COMMAND MODE.
0469 *
0470 STRTA XRA A
0471 MOV C,A
0472 LXI H,SYSRAM CLEAR THE FIRST PAGE
0473 *
0474 CLERA MOV M,A
0475 INX H

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

PAGE 5

```

C1B6 0C 0476 INR C
C1B7 C2 B4 C1 0477 JNZ CLERA
0478 *
0479 LXI SP,SYSTP SET UP THE STACK FOR CALL
C1BA 31 FF CB 0480 CALL PERSE
C1BD CD D5 C0 0481 COMN1 XRA A
C1C0 AF 0482 OUT STAPT BE SURE TAPES ARE OFF
C1C1 D3 FA 0483 STA OPORT
C1C3 32 07 C8 0484 STA IPORT
C1C6 32 06 C8 0485 *
0486 *
0487 *
0488 * --- COMMAND MODE ---
0489 *
0490 *
0491 * THIS ROUTINE GETS AND PROCESSES COMMANDS
0492 *
0493 COMND LXI SP,SYSTP SET STACK POINTER
C1C9 31 FF CB 0494 LDA OPORT GET PORT
C1CC 3A 07 C8 0495 PUSH PSW
C1CF F5 0496 XRA A
C1D0 AF 0497 STA OPORT FORCE SCREEN OPERATIONS
C1D1 32 07 C8 0498 CALL PROMPT PUT PROMPT ON SCREEN
C1D4 CD E1 C2 0499 CALL GCLIN GET COMMAND LINE
C1D7 CD E4 C1 0500 POP PSW
C1DA F1 0501 STA OPORT RESTORE DEFAULT PORT
C1DB 32 07 C8 0502 CALL COPRC PROCESS THE LINE
C1DE CD 05 C2 0503 JMP COMND OVER AND OVER
C1E1 C3 C9 C1 0504 *
0505 *
0506 *
0507 * -THIS ROUTINE READS A COMMAND LINE FROM THE SYSTEM
0508 * KEYBOARD
0509 *
0510 * C/R TERMINATES THE SEQUENCE ERASING ALL CHARS TO THE
0511 * RIGHT OF THE CURSOR
0512 * L/F TERMINATES THE SEQUENCE
0513 * MODE RESTARTS THE COMMAND LINE.
0514 *
0515 GCLIN CALL SINP READ INPUT DEVICE
C1E4 CD 1F C0 0516 JZ GCLIN
C1E7 CA E4 C1 0517 ANI 7FH CLEAR PARITY BIT
C1EA E6 7F 0518 JZ COMN1 THIS WAS A MODE (OR EVEN CTL-0)
C1EC CA C0 C1 0519 MOV B,A
C1EF 47 0520 CPI CR CARRIAGE RETURN
C1F0 FE 0D 0521 JZ CLINE YES--DONE WITH LINE
C1F2 CA F4 C0 0522 CPI LF LINE FEED
C1F5 FE 0A 0523 RZ YES--DONE WITH LINE, LEAVE AS IS
C1F7 C8 0524 CPI 7FH DELETE CHR?
C1F8 FE 7F 0525 JNZ CONT
C1FA C2 FF C1 0526 MVI B,BACKS REPLACE IT
C1FD 06 5F 0527 *
C1FF CD 19 C0 0528 CONT CALL SOUT

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977  
SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

```

C202 C3 E4 C1      0529      JMP      GCLIN
                   0530 *
                   0531 *
                   0532 *      FIND AND PROCESS COMMAND
                   0533 *
C205 CD 36 C1      0534 COPRC CALL  CREM  REMOVE THE CURSOR
C208 OE 01          0535      MVI    C,1   SET FOR CHARACTER POSITION
C20A CD 20 C1      0536      CALL   VDAD2  GET SCREEN ADDRESS
C20D EB            0537      XCHG
C20E 21 00 C0      0538      LXI     H,START MAKE SURE HL PT TO SOLOS START
C211 E5            0539      PUSH   H      SAVE IT FOR LATER DISPT
C212 CD 2E C3      0540      CALL   SCHR  SCAN PAST BLANKS
C215 CA 80 C4      0541      JZ     ERR1  NO COMMAND?
C218 EB            0542      XCHG    ..    HL HAS FIRST CHR
                   0543 *
C219 11 4A C2      0544      LXI     D,COMTAB POINT TO COMMAND TABLE
C21C CD 31 C2      0545      CALL   FDCOM  SEE IF IN PRIMARY COMMAND TABLE
C21F CC 2E C2      0546      CZ     FDCOU  IF NOT, TRY CUSTOM TABLE NEXT
                   0547 DISPO EQU    $      HERE TO SEE IF ERROR OR DISP
C222 CA 81 C4      0548      JZ     ERR2  NOT VALID, ERROR
C225 13            0549      INX     D      BUMP TO PTR OF RTN
C226 EB            0550      XCHG    .      HL PT TO RTN ADDR
                   0551 *
                   0552 *
                   0553 *      THIS IS THE DISPATCH ROUTINE.
                   0554 *      HL PT TO RTN ADDRESS, HL WILL BE RESTORED FM STACK
                   0555 *      SO THAT HL ARE RESTORED BEFORE DISPATCH.
                   0556 *
C227              0557 DISPT EQU    $      OFF TO A ROUTINE
C227 7E            0558      MOV     A,M    LO ADDR
C228 23            0559      INX     H
C229 66            0560      MOV     H,M    HI ADDR
C22A 6F            0561      MOV     L,A    HL NOW COMPLETE
C22B              0562 DISPT EQU    $      HERE TO GO OFF TO HL
C22B E3            0563      XTHL    .      XCHG HL W/HL ON STACK
C22C 7D            0564      MOV     A,L    ALSO COPY HERE FOR SETS
C22D C9            0565      RET     .      AND GO OFF TO THE RTN
                   0566 *
                   0567 *
                   0568 *      THIS ROUTINE SEARCHES THROUGH A TABLE, POINTED TO
                   0569 *      BY 'DE', FOR A DOUBLE CHARACTER MATCH OF THE 'HL'
                   0570 *      MEMORY CONTENT. IF NO MATCH IS FOUND THE SCAN ENDS
                   0571 *      WITH HL POINTING TO ORIGINAL VALUE AND ZERO FLAG SET.
                   0572 *
C22E 11 3C C8      0573 FDCOU LXI    D,CUTAB HERE TO SCAN CUSTOM TBL ONLY
                   0574 *
C231 1A            0575 FDCOM LDAX   D
C232 B7            0576      ORA     A      TEST FOR TABLE END
C233 C8            0577      RZ     .      NOT FOUND..COMMAND ERROR
C234 E5            0578      PUSH   H      SAVE START OF SCAN ADDRESS
C235 BE            0579      CMP     M      TEST FIRST CHR
C236 13            0580      INX     D
C237 C2 43 C2      0581      JNZ    NCOM

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977  
SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 6

```

C23A 23            0582 *
C23B 1A            0583      INX     H
C23C BE            0584      LDAX   D
C23D C2 43 C2      0585      CMP     M      NOW SECOND CHARACTER
                   0586      JNZ    NCOM  GOODNESS
                   0587 *
C240 E1            0588      POP     H      RESTORE ORIGINAL SCAN ADDR
C241 B7            0589      ORA     A      SET NON-ZERO FLAG SAYING FOUND
C242 C9            0590      RET     .      WITH NON-ZERO SET
                   0591 *
                   0592 *
C243 13            0593 NCOM INX     D      GO TO NEXT ENTRY
C244 13            0594      INX     D
C245 13            0595      INX     D
C246 E1            0596      POP     H      GET BACK ORIGINAL ADDRESS
C247 C3 31 C2      0597      JMP     FDCOM  CONTINUE SEARCH
                   0598 *
                   0599 *
                   0600 *      COMMAND TABLE
                   0601 *
                   0602 *      THIS TABLE DESCRIBES THE VALID COMMANDS FOR SOLOS
                   0603 *
C24A 54 45          0604 COMTAB ASC  'TE'  TERMINAL MODE
C24C 67 C3          0605      DW     TERM
C24E 44 55          0606      ASC    'DU'  DUMP
C250 BF C3          0607      DW     DUMP
C252 45 4E          0608      ASC    'EN'  ENTR
C254 23 C4          0609      DW     ENTER
C256 45 58          0610      ASC    'EX'  EXEC
C258 5E C4          0611      DW     EXEC
C25A 47 45          0612      ASC    'GE'  GET A FILE
C25C A7 C4          0613      DW     TLOAD
C25E 53 41          0614      ASC    'SA'  SAVE A FILE
C260 E6 C4          0615      DW     TSAVE
C262 58 45          0616      ASC    'XE'  XEQ (EXECUTE) A FILE
C264 A6 C4          0617      DW     TXEQ
C266 43 41          0618      ASC    'CA'  CATALOG OF FILES
C268 2B C5          0619      DW     TLIST
C26A 53 45          0620      ASC    'SE'  SET COMMAND
C26C 7A C5          0621      DW     SET
C26E 43 55          0622      ASC    'CU'  CUSTOM COMMAND
C270 BD C5          0623      DW     CUSET
C272 00            0624      DB     0      END OF TABLE MARK
                   0625 *
                   0626 *
                   0627 *      DISPLAY DRIVER COMMAND TABLE
                   0628 *
                   0629 *      THIS TABLE DEFINES THE CHARACTERS FOR SPECIAL
                   0630 *      PROCESSING. IF THE CHARACTER IS NOT IN THE TABLE IT
                   0631 *      GOES TO THE SCREEN.
                   0632 *
C273 0B            0633 TBL   DB     CLEAR-80H  SCREEN
C274 D5 C0          0634      DW     PERSE

```

PROGRAM DEVELOPMENT SYSTEM

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

```

C276 17      0635      DB      UP-80H  CURSOR
C277 04 C1    0636      DW      PUP
C279 1A      0637      DB      DOWN-80H
C27A CB C0    0638      DW      PDOWN
C27C 01      0639      DB      LEFT-80H
C27D 0B C1    0640      DW      PLEFT
C27F 13      0641      DB      RIGHT-80H
C280 15 C1    0642      DW      PRIT
C282 0E      0643      DB      HOME-80H
C283 E5 C0    0644      DW      PHOME
C285 0D      0645      DB      CR      CARRIAGE RETURN
C286 47 C1    0646      DW      PCR
C288 0A      0647      DB      LF      LINE FEED
C289 4D C1    0648      DW      PLF
C28B 5F      0649      DB      BACKS  BACK SPACE
C28C 3E C1    0650      DW      PBACK
C28E 1B      0651      DB      ESC    ESCAPE KEY
C28F 59 C1    0652      DW      PESC
C291 00      0653      DB      0      END OF TABLE
0654 *
0655 *
0656 *      OUTPUT DEVICE TABLE
0657 *
C292 54 C0    0658 OTAB  DW      VDMOT  VDM DRIVER
C294 4A C0    0659      DW      SDROT  SERIAL OUTPUT
C296 E6 C2    0660      DW      PROUT  PARALLEL OUTPUT
C298 D2 C2    0661      DW      ERROT  ERROR OR USER DRIVER HANDLER
0662 *
0663 *
0664 *      INPUT DEVICE TABLE
0665 *
C29A 2E C0    0666 ITAB  DW      KSTAT  KEYBOARD INPUT
C29C 42 C0    0667      DW      SSTAT  SERIAL INPUT
C29E DD C2    0668      DW      PASTAT  PARALLEL INPUT
C2A0 CB C2    0669      DW      ERRIT  ERROR OR USER DRIVER HANDLER
0670 *
0671 *
0672 *      SECONDARY COMMAND TABLE FOR SET COMMAND
0673 *
C2A2 54 41    0674 SETAB ASC    'TA'   SET TAPE SPEED
C2A4 8E C5    0675      DW      TASPD
C2A6 53 3D    0676      ASC    'S='   SET DISPLAY SPEED
C2A8 99 C5    0677      DW      DISPD
C2AA 49 3D    0678      ASC    'I='   SET INPUT PORT
C2AC 9D C5    0679      DW      SETIN
C2AE 4F 3D    0680      ASC    'O='   SET OUTPUT PORT
C2B0 A1 C5    0681      DW      SETOT
C2B2 4E 3D    0682      ASC    'N='   NULLS
C2B4 B5 C5    0683      DW      SETNU
C2B6 43 49    0684      ASC    'CI'   SET CUSTOM DRIVER ADDRESS
C2B8 A5 C5    0685      DW      SETCI
C2BA 43 4F    0686      ASC    'CO'   SET CUSTOM OUTPUT DRIVER ADDRESS
C2BC A9 C5    0687      DW      SETCO

```

PROGRAM DEVELOPMENT SYSTEM

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 7

```

C2BE 58 45    0688      ASC    'XE'   SET HEADER XEQ ADDRESS
C2C0 B1 C5    0689      DW      SETXQ
C2C2 54 59    0690      ASC    'TY'   SET HEADER TYPE
C2C4 AD C5    0691      DW      SETTY
C2C6 43 52    0692      ASC    'CR'   SET CRC TO ALLOW IGNORING OF CRC ERRORS
C2C8 B9 C5    0693      DW      SETCR
C2CA 00      0694      DB      0      END OF TABLE MARK
0695 *
0696 *
0697 *      SOLOS PORT ERROR HANDLER
0698 *
C2CB E5      0699 ERRIT  PUSH  H      SAVE HL ONCE AGAIN
C2CC 2A 00 C8 0700      LHL  UIPRT  GET USER INPUT PORT ADDRESS
C2CF C3 D6 C2 0701      JMP   ERROT  AND GO PROCESS
0702 *
C2D2 E5      0703 ERROT  PUSH  H
C2D3 2A 02 C8 0704      LHL  UOPRT  GET USER OUTPUT PORT ADDRESS
C2D6 7D      0705 ERROT  MOV   A,L    TEST HL FOR ZERO
C2D7 B4      0706      ORA   H
C2D8 CA C0 C1 0707      JZ    COMN1  IF ZERO RETURN TO COMMAND MODE
C2DB E3      0708      XTHL  .      ADDRESS TO STACK...OLD HL TO HL
C2DC C9      0709      RET   .      GO TO THE DRIVER
0710 * --
9999      COPY  SOLOS2/1
0711 *
0712 *      THIS ROUTINE IS THE PARALLEL DEVICE HANDLER
0713 *      NO PROVISION IS MADE FOR CONTROLLING THE PORT
0714 *      CONTROL BIT.
0715 *
0716 *      PARALLEL INPUT DRIVER
0717 *
C2DD DB FA    0718 *
C2DF 2F      0719 PASTAT IN      STAPT
C2E0 E6 02    0720      CMA      .      INVERT STATUS FLAGS
C2E2 C8      0721      ANI      PDR  .      TEST BIT
C2E3 DB FD    0722      RZ      .      WITH FLAG SET
C2E5 C9      0723      IN      PDATA  GET DATA
0724      RET
0725 *
0726 *      PARALLEL OUTPUT HANDLER
0727 *
C2E6 DB FA    0728 PROUT  IN      STAPT  GET STATUS
C2E8 E6 04    0729      ANI      PXDR  TEST IF DEVICE IS READY
C2EA C2 E6 C2 0730      JNZ      PROUT  LOOP UNTIL SO
C2ED 78      0731      MOV      A,B
C2EE D3 FD    0732      OUT      PDATA
C2F0 C9      0733      RET
0734 *
0735 *
0736 *      OUTPUT A CRLF FOLLOWED BY A PROMPT
0737 *
C2F1 CD F9 C2 0738 PROMPT CALL  CRLF
C2F4 06 3E    0739      MVI      B,>>  THE PROMPT

```

2 OF 3 \*\*\*\*\*

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

C2F6	C3	19	C0	2740	JMP	SOUT	PUT IT ON THE SCREEN
				2741	*		
				2742	*		
C2F9	06	0A		2743	CRLF	MVI B,LF	LINE FEED
C2FB	CD	19	C0	2744		CALL SOUT	
C2FE	06	0D		2745		MVI B,'CR	CARRIAGE RETURN
C300	CD	19	C0	2746		CALL SOUT	
				2747	* NOW	OUTPUT THE NULLS	
C303	3A	10	C8	2748	LDA	NUCNT	GET DESIRED COUNT
C306	4F			2749	MOV	C,A	STORE IN C
C307	0D			2750	NULOT	DCR C	
C308	F8			2751	RM	.	RETURN WHEN PAST ZERO
C309	AF			2752	XRA	A	GET A NULL
C30A	CD	1F	C4	2753	CALL	OUTH	
C30D	C3	07	C3	2754	JMP	NULOT	
				2755	*		
				2756	*		
				2757	*	SCAN OFF	OPTIONAL PARAMETER. IF PRESENT RETURN WITH
				2758	*	VALUE IN HL	AND COPY OF "L" IN "A". IF NOT PRESENT
				2759	*	RETURN WITH A "1"	IN "A" AND HL UNTOUCHED.
				2760	*		
C310	CD	1B	C3	2761	PSCAN	CALL SBLK	
C313	3E	01		2762		MVI A,1	DEFAULT VALUE
C315	C8			2763		RZ	IF NONE
C316	CD	40	C3	2764		CALL SHEX	CONVERT VALUE
C319	7D			2765		MOV A,L	GET LOWER HALF
C31A	C9			2766		RET	
				2767	*		
				2768	*		
				2769	*	SCAN OVER UP TO 12	CHARACTERS LOOKING FOR A BLANK
				2770	*		
C31B	0E	0C		2771	SBLK	MVI C,12	MAXIMUM COMMAND STRING
C31D	1A			2772	SBLK1	LDAX D	
C31E	FE	20		2773		CPI BLANK	
C320	CA	2E	C3	2774		JZ SCHR	GOT A BLANK NOW SCAN PAST IT
C323	13			2775		INX D	
C324	FE	3D		2776		CPI '='	ALSO ALLOW AN EQUAL TO STOP US
C326	CA	2E	C3	2777		JZ SCHR	IF SO, PTR AT CHAR FOLLOWING
C329	0D			2778		DCR C	NO MORE THAN TWELVE
C32A	C2	1D	C3	2779		JNZ SBLK1	
C32D	C9			2780		RET	GO BACK WITH ZERO FLAG SET
				2781	*		
				2782	*		
				2783	*	SCAN PAST UP TO 10	BLANK POSITIONS LOOKING FOR
				2784	*	A NON BLANK CHARACTER.	
				2785	*		
C32E	0E	0A		2786	SCHR	MVI C,10	SCAN TO FIRST NON BLANK CHR WITHIN 10
C330	1A			2787	SCHR1	LDAX D	GET NEXT CHARACTER
C331	FE	20		2788		CPI SPACE	
C333	C0			2789		RNZ	WE'RE PAST THEM
C334	13			2790		INX D	NEXT SCAN ADDRESS
C335	0D			2791		DCR C	
C336	C8			2792		RZ	COMMAND ERROR

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

```
C337 C3 30 C3          2793 JMP      SCHR1    KEEP LOOPING
                        2794 *
                        2795 *
                        2796 * THIS ROUTINE SCANS OVER CHARACTERS, PAST BLANKS AND
                        2797 * CONVERTS THE FOLLOWING VALUE TO HEX. ERRORS RETURN TO
                        2798 * THE ERROR HANDLER.
                        2799 *
C33A CD 1B C3          2800 SCONV CALL   SBLK    FIND IF VALUE IS PRESENT
C33D CA 80 C4          2801         JZ     ERR1    ABORT TO ERROR IF NONE
                        2802 *
                        2803 *
                        2804 * THIS ROUTINE CONVERTS ASCII DIGITS INTO BINARY FOLLOWING
                        2805 * A STANDARD HEX CONVERSION. THE SCAN STOPS WHEN AN ASCII
                        2806 * SPACE IS ENCOUNTERED. PARAMETER ERRORS REPLACE THE ERROR
                        2807 * CHARACTER ON THE SCREEN WITH A QUESTION MARK.
                        2808 *
C340 21 00 00          2809 SHEX LXI     H,0      CLEAR H & L
C343 1A                2810 SHE1 LDAX   D        GET CHARACTER
C344 FE 20             2811 CPI     20H     IS IT A SPACE?
C346 C8               2812 RZ              IF SO
C347 FE 2F             2813 CPI     '/'    SLASH IS ALSO LEGAL
C349 C8               2814 RZ              IF NOT
C34A FE 3A             2815 CPI     ':'    EVEN THE COLON IS ALLOWED
C34C C8               2816 RZ              IF NOT
                        2817 *
C34D 29               2818 HCONV DAD     H        MAKE ROOM FOR THE NEW ONE
C34E 29               2819 DAD     H
C34F 29               2820 DAD     H
C350 29               2821 DAD     H
C351 CD 5D C3          2822 CALL   HCOV1   DO THE CONVERSION
C354 D2 80 C4          2823 JNC     ERR1   NOT VALID HEXIDECEMAL VALUE
C357 85               2824 ADD     L
C358 6F               2825 MOV     L,A     MOVE IT IN
C359 13               2826 INX     D        BUMP THE POINTER
C35A C3 43 C3          2827 JMP     SHE1
                        2828 *
C35D D6 30             2829 HCOV1 SUI     48     REMOVE ASCII BIAS
C35F FE 0A             2830 CPI     10
C361 D8               2831 RC              IF LESS THAN 9
C362 D6 07             2832 SUI     7       IT'S A LETTER
C364 FE 10             2833 CPI     10H
C366 C9               2834 RET              WITH TEST IN HAND
                        2835 *
                        2836 *
                        2837 *
                        2838 * TERM COMMAND
                        2839 *
                        2839 * THIS ROUTINE GETS CHARACTERS FROM THE SYSTEM KEYBOARD
                        2840 * AND OUTPUTS THEM TO THE SELECTED OUTPUT PORT. IT IS
                        2841 * INTENDED TO CONFIGURE THE Sol AS A STANDARD VIDEO
                        2842 * TERMINAL. COMMAND KEYS ARE NOT OUTPUT TO THE OUTPUT
                        2843 * PORT BUT ARE INTERPRETED AS DIRECT Sol COMMANDS.
                        2844 * THE MODE COMMAND, RECEIVED BY THE KEYBOARD, PUTS THE
                        2845 * Sol IN THE COMMAND MODE.
```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

```

0846 *
0847 *
0848 *
0849 TERM CALL PSCAN FIND IF INPUT PARAMETER IS PRESENT
0850 STA IPORT SINP WILL USE THIS DRIVER (DEFAULT IS 1)
0851 CALL PSCAN NOW FOR THE OUTPUT DRIVER
0852 STA OPORT
0853 *
0854 TERM1 CALL KSTAT IS THERE ONE WAITING?
0855 JZ TIN IF NOT
0856 MOV B,A SAVE IT IN B
0857 CPI MODE IS IT MODE
0858 JZ COMN1 YES--RESET AND QUIT TERM
0859 JC TOUT NON-CURSOR KEY---SEND TO TERM PORT
0860 CALL VDMOT PROCESS IT
0861 JMP TIN
0862 *
0863 TOUT CALL SOUT OUTPUT IT TO THE SERIAL PORT
0864 TIN CALL SINP GET INPUT STATUS
0865 JZ TERM1 LOOP IF NOT
0866 ANI 7FH NO HIGH BITS FROM HERE
0867 JZ TERM1 A NULL IS IGNORED
0868 MOV B,A IT'S OUTPUT FROM 'B'
0869 CPI 1BH IS IT A CONTROL CHAR TO BE IGNORED
0870 JNC TERM2 NO--TO VDM AS IS THEN
0871 CPI CR CR OR LF ARE SPECIAL CASES THOUGH
0872 JZ TERM2 AND MUST BE PASSED STD MODE TO VDM DRIVER
0873 CPI LF
0874 JZ TERM2
0875 LDA ESCFL A CTL CHAR---ARE WE W/IN AN ESC SEQUENCE?
0876 ORA A IF YES, THEN OUTPUT CTL CHAR DIRECTLY TO VDM
0877 JNZ TERM2 WE SURE ARE, LET VDM DRIVER HANDLE IT
0878 PUSH B SAVE THE CHAR
0879 MVI B,ESC CTL CHAR TO VDM VIA ESC SEQUENCE
0880 CALL VDMOT
0881 MVI B,7 SAY TO PUT OUT NEXT CHAR AS IS
0882 CALL VDMOT ALMOST READY
0883 POP B RESTORE CHAR
0884 TERM2 EQU $ ALL READY TO OUTPUT THE CHAR
0885 CALL VDMOT PUT IT ON THE SCREEN
0886 JMP TERM1 LOOP OVER AND OVER
0887 *
0888 *
0889 *
0890 * DUMP COMMAND
0891 *
0892 * THIS ROUTINE DUMPS CHARACTERS FROM MEMORY TO THE
0893 * CURRENT OUTPUT DEVICE. ALL VALUES ARE DISPLAYED AS
0894 * ASCII HEX.
0895 *
0896 * THE COMMAND FORM IS AS FOLLOWS:
0897 *
0898 * Dump addr1 addr2

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 9

```

0899 *
0900 * THE VALUES FROM ADDR1 TO ADDR2 ARE THEN OUTPUT TO THE
0901 * OUTPUT DEVICE. IF ONLY ADDR1 IS SPECIFIED THEN THE
0902 * VALUE AT THAT ADDRESS IS OUTPUT.
0903 *
0904 DUMP CALL SCONV SCAN TO FIRST ADDRESS AND CONVERT IT
0905 PUSH H SAVE THE VALUE
0906 CALL PSCAN SEE IF SECOND WAS GIVEN
0907 POP D GET BACK START
0908 XCHG HL HAS START, DE HAS END
0909 *
0910 DLOOP CALL CRLF
0911 CALL ADOUT OUTPUT ADDRESS
0912 CALL BOUT ANOTHER SPACE TO KEEP IT PRETTY
0913 MVI C,16 VALUES PER LINE
0914 *
0915 DLP1 MOV A,M GET THE CHR
0916 PUSH B SAVE VALUE COUNT
0917 CALL HBOUT SEND IT OUT WITH A BLANK
0918 MOV A,L COMPARE DE & HL
0919 SUB E
0920 MOV A,H
0921 SBB D
0922 JNC COMND ALL DONE
0923 POP B VALUES PER LINE
0924 INX H
0925 DCR C BUMP THE LINE COUNT
0926 JNZ DLP1 NOT ZERO IF MORE FOR THIS LINE
0927 JMP DLOOP DO A LPCR BEFORE THE NEXT
0928 *
0929 *
0930 * OUTPUT HL AS HEX 16 BIT VALUE
0931 *
0932 ADOUT MOV A,H H FIRST
0933 CALL HEOUT
0934 MOV A,L THEN "L" FOLLOWED BY A SPACE
0935 *
0936 HBOUT CALL HEOUT
0937 CALL SINP SEE IF A CHAR WAITING
0938 JZ BOUT NO
0939 ANI 7FH CLR PARITY 1ST THO
0940 JZ COMND EITHER MODE OR CTL-@
0941 CPI IS IT A SPACE
0942 JNZ BOUT NO--IGN THE CHAR
0943 WTLP1 CALL SINP IF SPACE, WAIT UNTIL ANY OTHER KEY HIT
0944 JZ WTLP1 THIS ALLOWS LOOKING AT THE DISPLAY
0945 BOUT MVI B,
0946 JMP SOUT PUT IT OUT
0947 *
0948 HEOUT MOV C,A GET THE CHARACTER
0949 RRC
0950 RRC MOVE THE HIGH FOUR DOWN
0951 RRC

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

```

C40F 0F          0952      RRC
C410 CD 14 C4    0953      CALL HEOU1 PUT THEM OUT
C413 79          0954      MOV A,C THIS TIME THE LOW FOUR
                  0955 *
C414 E6 0F      0956 HEOU1 ANI OFH FOUR ON THE FLOOR
C416 C6 30      0957      ADI 48 WE WORK WITH ASCII HERE
C418 FE 3A      0958      CPI 58 0-9?
C41A DA 1F C4   0959      JC OUTH YUP!
C41D C6 07      0960      ADI 7 MAKE IT A LETTER
C41F 47          0961 OUTH MOV B,A OUTPUT IT FROM REGISTER 'B'
C420 C3 19 C0   0962      JMP SOUT
                  0963 *
                  0964 *
                  0965 *
                  0966 *
                  0967 *
                  0968 *
                  0969 *
                  0970 *
                  0971 *
                  0972 *
                  0973 *
                  0974 *
                  0975 *
                  0976 *
                  0977 *
                  0978 *
                  0979 *
                  0980 *
                  0981 *
                  0982 *
                  0983 *
                  0984 *
                  0985 *
                  0986 *
                  0987 *
                  0988 *
                  0989 *
                  0990 *
                  0991 *
                  0992 *
                  0993 *
                  0994 *
                  0995 *
                  0996 *
                  0997 *
                  0998 *
                  0999 *
                  1000 *
                  1001 *
                  1002 *
                  1003 *
                  1004 *
C423 CD 3A C3   0975 ENTER CALL SCONV SCAN OVER CHARS AND GET ADDRESS
C426 E5          0976      PUSH H SAVE ADDRESS
C427 AF          0977      XRA A
C428 32 07 C8   0978      STA OPORT ENTER VALUES TO SCREEN BUFFER
                  0979 *
C42B CD F9 C2   0980 ENLOP CALL CRLF
C42E 06 3A      0981      MVI B, ' '
C430 CD FF C1   0982      CALL CONT GET LINE OF INPUT
C433 CD 36 C1   0983      CALL CREM REMOVE THE CURSOR
C436 0E 01      0984      MVI C,1 START SCAN
C438 CD 20 C1   0985      CALL VDAD2 GET ADDRESS
C43B EB          0986      XCHG . ....TO DE
                  0987 *
                  0988 *
C43C 0E 03      0989 ENLO1 MVI C,3 NO MORE THAN THREE SPACES BETWEEN VALUES
C43E CD 30 C3   0990      CALL SCHR1 SCAN TO NEXT VALUE
C441 CA 2B C4   0991      JZ ENLOP LAST ENTRY FOUND START NEW LINE
                  0992 *
C444 FE 2F      0993      CPI '/' COMMAND TERMINATOR?
C446 CA C0 C1   0994      JZ COMN1 IF SO...RETURN TO STANDARD INPUT
C449 CD 40 C3   0995      CALL SHEX CONVERT VALUE
C44C FE 3A      0996      CPI ':' ADDRESS TERMINATOR?
C44E CA 59 C4   0997      JZ ENLO3 GO PROCESS IF SO
C451 7D          0998      MOV A,L GET LOW PART AS CONVERTED
C452 E1          0999      POP H GET MEMORY ADDRESS
C453 77          1000      MOV M,A PUT IN THE VALUE
C454 23          1001      INX H
C455 E5          1002      PUSH H
C456 C3 3C C4   1003      JMP ENLO1 CONTINUE THE SCAN
                  1004 *

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 10

```

C459 E3          1005 ENLO3 XTHL . PUT NEW ADDRESS ON STACK
C45A 13          1006      INX D MOVE SCAN PAST TERMINATOR
C45B C3 3C C4   1007      JMP ENLO1
                  1008 *
                  1009 *
                  1010 *
                  1011 *
                  1012 *
                  1013 *
                  1014 *
                  1015 *
                  1016 *
                  1017 *
                  1018 *
                  1019 *
                  1020 *
                  1021 *
                  1022 *
                  1023 *
                  1024 *
                  1025 *
                  1026 *
                  1027 *
                  1028 *
                  1029 *
                  1030 *
                  1031 *
                  1032 *
                  1033 *
                  1034 *
                  1035 *
                  1036 *
                  1037 *
                  1038 *
                  1039 *
                  1040 *
                  1041 *
                  1042 *
                  1043 *
                  1044 *
                  1045 *
                  1046 *
                  1047 *
                  1048 *
                  1049 *
                  1050 *
                  1051 *
                  1052 *
                  1053 *
                  1054 *
                  1055 *
                  1056 *
                  1057 *
C45E CD 3A C3   1021 EXEC CALL SCONV SCAN PAST BLANKS AND GET PARAMETER
C461 E5          1022 EXEC1 PUSH H PUT GO ADDRESS ON STACK
C462 21 00 C0   1023      LXI H,START TELL THE PROGRAM WHERE WE CAME FROM
C465 C9          1024      RET AND DISPATCH TO IT
                  1025 *
                  1026 *
                  1027 *
                  1028 *
                  1029 *
                  1030 *
                  1031 *
                  1032 *
                  1033 *
                  1034 *
                  1035 *
                  1036 *
                  1037 *
                  1038 *
                  1039 *
                  1040 *
                  1041 *
                  1042 *
                  1043 *
                  1044 *
                  1045 *
                  1046 *
                  1047 *
                  1048 *
                  1049 *
                  1050 *
                  1051 *
                  1052 *
                  1053 *
                  1054 *
                  1055 *
                  1056 *
                  1057 *
C466 21 1C C8   1033 NAMES LXI H,THEAD POINT TO INTERNAL HEADER
C469 CD 1B C3   1034 NAME CALL SBLK SCAN OVER TO FIRST CHRS
C46C 06 06      1035      MVI B,6 UP TO SIX ARE ACCEPTED
                  1036 *
C46E 1A          1037 NAME1 LDAX D GET CHARACTER
C46F FE 20      1038      CPI NO UNIT DELIMITER
C471 CA 86 C4   1039      JZ NFIL
C474 FE 2F      1040      CPI '/' UNIT DELIMITER
C476 CA 86 C4   1041      JZ NFIL
C479 77          1042      MOV M,A
C47A 13          1043      INX D BUMP THE SCAN POINTER
C47B 23          1044      INX H
C47C 05          1045      DCR B
C47D C2 6E C4   1046      JNZ NAME1 FALL THROUGH TO ERR1 IF TOO MANY CHRS IN NAME
                  1047 *
                  1048 *
                  1049 *
                  1050 *
                  1051 *
                  1052 *
                  1053 *
                  1054 *
                  1055 *
                  1056 *
                  1057 *
C480 EB          1051 ERR1 XCHG . GET SCAN ADDRESS TO HL
C481 36 3F      1052 ERR2 MVI M, '?' PUT QUESTION MARK ON SCREEN
C483 C3 C0 C1   1053      JMP COMN1 AND RETURN TO COMMAND MODE
                  1054 *
                  1055 *
                  1056 *
                  1057 *

```



\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

```

1058 *
1059 NFIL MVI M,0 PUT IN AT LEAST ONE ZERO
1060 INX H
1061 DCR B
1062 JNZ NFIL LOOP UNTIL B IS ZERO
1063 *
1064 CPI '/' IS THERE A UNIT SPECIFICATION?
1065 MVI A,1 PRETEND NOT
1066 JNZ DEFLT
1067 INX D MOVE PAST THE TERMINATOR
1068 CALL SCHR GO GET UNIT SPEC
1069 SUI '0' REMOVE ASCII BIAS
1070 *
1071 DEFLT EQU $ MOVE OVER TO INTERNAL REPRESENTATION
1072 ANI 1 JUST BIT ZERO
1073 MVI A,TAPE1 ASSUME TAPE ONE
1074 JNZ STUNT IF NON-ZERO, ITS ONE
1075 RAR
1076 STUNT STA FNUMF SET IT IN
1077 RET
1078 *
1079 *
1080 *
1081 * THIS ROUTINE PROCESSES THE XEO AND GET COMMANDS
1082 *
1083 *
1084 TXEQ DB 3EH THIS BEGINS "MVI A,0AFH"
1085 TLOAD XRA A A=0 MEANS TLOAD, ELSE TXEQ
1086 PUSH PSW SAVE FLAG FOR LATER
1087 LXI H,DHEAD PLACE DUMMY HEADER HERE
1088 CALL NAME SET IN NAME AND UNIT
1089 LXI H,0 PRETEND NO SECOND VALUE
1090 CALL PSCAN GO GET THE ADDRESS (IF PRESENT)
1091 *
1092 TLOA2 XCHG . PUT ADDRESS IN DE
1093 LXI H,DHEAD PT TO DUMMY HEADER W/ NAME TO LOAD
1094 MOV A,M SEE IF A NAME WAS ENTERED
1095 ORA A IS THERE A NAME?
1096 JNZ TLOA3 YES--SEARCH FOR IT
1097 LXI H,THEAD NO NAME, LOAD 1ST FILE
1098 TLOA3 PUSH H SAVE PTR TO NAME TO LOAD
1099 CALL ALOAD GET UNIT AND SPEED
1100 POP H RESTORE PTR TO HDR TO LOAD
1101 CALL RTAPE READ IN THE TAPE
1102 JC TAERR TAPE ERROR?
1103 *
1104 CALL NAOUT PUT OUT THE HEADER PARAMETERS
1105 POP PSW RESTORE FLAG FROM ORIGINAL ENTRY
1106 ORA A
1107 RZ . AUTO XEO NOT WANTED
1108 LDA HTYPE CHECK TYPE
1109 ORA A SET FLAGS
1110 JM TAERR TYPE IS NON XEO

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 11

```

1111 LDA THEAD+5 GET CHARACTER PAST NAME
1112 ORA A
1113 JNZ TAERR THE BYTE MUST BE ZERO FOR AUTO XEO
1114 LHLD XEQAD GET THE TAPE ADDRESS
1115 JMP EXEC1 AND GO TO IT
1116 *
1117 *
1118 * -- GET --
1119 *
1120 * THIS ROUTINE IS USED TO SAVE PROGRAMS AND DATA ON
1121 * THE CASSETTE UNIT.
1122 *
1123 *
1124 TSAVE CALL NAMES GET NAME AND UNIT
1125 CALL SCONV GET START ADDRESS
1126 PUSH H USE THE STACK AS A REGISTER
1127 CALL SCONV GET END ADDRESS
1128 XTHL . PUT END ON STACK, GET BACK START
1129 PUSH H SAVE START ON TOP OF STACK
1130 CALL PSCAN SEE IF OPTIONAL HEADER ADDRESS WAS GIVEN
1131 SHLD LOADR PUT HEADER ADDRESS IN PLACE
1132 *
1133 POP H "FROM" ADDRESS TO HL
1134 POP D GET BACK "END" ADDRESS
1135 PUSH H SAVE FROM AGAIN FOR LATER
1136 MOV A,E NOW CALCULATE SIZE
1137 SUB L SIZE=END-START+1
1138 MOV L,A
1139 MOV A,D
1140 SBB H
1141 MOV H,A
1142 INX H
1143 SHLD BLOCK STORE THE SIZE
1144 PUSH H SAVE IT FOR THE READ ALSO
1145 *
1146 CALL ALOAD GET UNIT AND SPEED
1147 LXI H,THEAD POINT TO HEADER
1148 CALL WHEAD AND WRITE IT OUT
1149 * NOW WRITE OUT THE DATA
1150 POP D GET SIZE TO DE
1151 POP H GET BACK "FROM" ADDRESS
1152 JMP WRLO1 WRITE OUT THE DATA AND RETURN
1153 *
1154 *
1155 * OUTPUT ERROR AND HEADER
1156 *
1157 TAERR CALL CRLF
1158 MVI D,6
1159 LXI H,ERRM POINT TO ERROR MESSAGE
1160 CALL NLOOP OUTPUT ERROR
1161 CALL NAOUT THEN THE HEADER
1162 JMP COMN1 AND BE SURE THE TAPE UNITS ARE OFF
1163 *

```

PROGRAM DEVELOPMENT SYSTEM

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

```

C525 45 52 52 4F 1164 ERRM ASC !ERROR !
      52 20
1165 *
1166 *
1167 * THIS ROUTINE READS HEADERS FROM THE TAPE AND OUTPUTS
1168 * THEM TO THE OUTPUT DEVICE. IT CONTINUES UNTIL THE
1169 * MODE KEY IS DEPRESSED.
1170 *
1171 TLIST CALL NAMES SET UP UNIT IF GIVEN
1172 CALL CRLF
1173 *
1174 *
1175 LLIST CALL ALOAD
1176 MVI B,1
1177 CALL TON TURN ON THE TAPE
1178 *
1179 LIST1 CALL RHEAD
1180 JC COMN1 TURN OFF THE TAPE UNIT
1181 JNZ LIST1
1182 CALL NAOUT OUTPUT THE HEADER
1183 JMP LIST1 LOOP UNTIL MODE IS DEPRESSED
1184 *
1185 *
1186 * THIS ROUTINE GETS THE CASSETTE UNIT NUMBER AND
1187 * SPEED TO REGISTER "A" FOR THE TAPE CALLS
1188 *
1189 ALOAD LXI H,FNUMF POINT TO THE UNIT SPECIFICATION
1190 LDA TSPD GET THE TAPE SPEED
1191 ORA M PUT THEM TOGETHER
1192 RET AND GO BACK
1193 *
1194 *
1195 * THIS ROUTINE OUTPUTS THE NAME AND PARAMETERS OF
1196 * THEAD TO THE OUTPUT DEVICE.
1197 *
1198 *
1199 NAOUT MVI D,8
1200 LXI H,THEAD-1 POINT TO THE HEADER
1201 CALL NLOOP OUTPUT THE HEADER
1202 CALL BOUT ANOTHER BLANK
1203 LHL D LOADR NOW THE LOAD ADDRESS
1204 CALL ADOUT PUT IT OUT
1205 LHL D BLOCK AND THE BLOCK SIZE
1206 CALL ADOUT
1207 JMP CRLF DO THE CRLF AND RETURN
1208 *
1209 *
1210 NLOOP MOV A,M GET CHARACTER
1211 ORA A
1212 JNZ CHR LI IF IT ISN'T A ZERO
1213 MVI A,
1214 CHR LI CALL OUTH OUTPUT CHAR NOW
1215 INX H

```

PROGRAM DEVELOPMENT SYSTEM

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 12

```

C575 15 1216 DCR D
C576 C2 6A C5 1217 JNZ NLOOP
C579 C9 1218 RET
1219 *
1220 *
1221 *
1222 *
1223 * "SET" COMMAND
1224 *
1225 * THIS ROUTINE GETS THE ASSOCIATED PARAMETER AND
1226 * DISPATCHES TO THE PROPER ROUTINE FOR SETTING
1227 * GLOBAL VALUES.
1228 *
1229 SET EQU $ THIS IS THE SET COMMAND
1230 CALL SBLK LOOK FOR SET NAME
1231 JZ ERR1 MUST HAVE AT LEAST SOMETHING!!
1232 PUSH D SAVE SCAN ADDRESS
1233 CALL SCONV CONVERT FOLLOWING VALUE
1234 XTHL GET SCAN ADDRESS BACK..SAVE VALUE ON STACK
1235 LXI D,SETAB SECONDARY COMMAND TABLE
1236 CALL FDCOM SEE IF IN TABLE
1237 JMP DISPO AND EITHER ERR OR OFF TO IT
1238 *
1239 *
1240 * THIS ROUTINE SETS THE TAPE SPEED
1241 *
1242 TASP D ORA A IS IT ZERO?
1243 JZ SETSP YES--THAT'S A VALID SPEED
1244 MVI A,32 SET TO SLOW IF NON-ZERO
1245 SETSP STA TSPD SPEED IS STORED HERE
1246 RET
1247 *
1248 *
1249 STSP D MOV A,B ESCAPE COMES HERE TO SET SPEED
1250 DISPD STA SPEED SET DISPLAY SPEED
1251 RET
1252 *
1253 * SET INPUT DRIVER
1254 *
1255 SETIN EQU $
1256 STA IPORT
1257 RET
1258 *
1259 * SET OUTPUT DRIVER
1260 *
1261 SETOT EQU $
1262 STA OPORT
1263 RET
1264 *
1265 * SET USERS CUSTOM INPUT DRIVER ADDRESS
1266 *
1267 SETCI SHLD UIPRT
1268 RET

```

```

**      PROGRAM DEVELOPMENT SYSTEM      **

SOLOS(TM) 77-03-27
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.
P.O. BOX 5260
SAN MATEO, CA 94402

1269 *
1270 * SET USERS CUSTOM OUTPUT DRIVER ADDRESS
1271 *
1272 SETCO SHLD UOPRT
1273 RET
1274 *
1275 * SET TYPE BYTE INTO HEADER
1276 *
1277 SETTY STA HTYPE
1278 RET
1279 *
1280 * SET EXECUTE ADDRESS INTO HEADER
1281 *
1282 SETXQ SHLD XEQAD
1283 RET
1284 *
1285 *
1286 SETNU STA NUCNT SET THE NULL COUNT
1287 RET . THAT'S DONE
1288 *
1289 *
1290 SETCR EQU $ SET TO IGNORE CRC ERRORS
1291 STA IGNCR FF=IGNORE ERRORS, ELSE=NORMAL
1292 RET .
1293 *
1294 *
1295 *
1296 * CUSTOM COMMAND NAME AND ADDRESS INTO CUSTOM COMMAND
1297 *
1298 CUSET CALL NAMES CUSTOM COMMAND ENTRY/REMOVAL
1299 LXI H,COMND DEFAULT ADDR IF NONE GIVEN
1300 CALL PSCAN GET RTN ADDR
1301 PUSH H SAVE RTN ADDR
1302 LXI H,THEAD PT AT NAME TO SEARCH
1303 CALL FDCOV SEARCH IT IN CUSTOM TABLE
1304 JZ CUSE2 NOT IN TABLE--ENTER IT
1305 DCX D IN TABLE, REMOVE IT
1306 MVI M,0 CHANGE NEW NAME TO BE ZERO
1307 CUSE2 MOV A,M GET 1ST CHAR OF NAME
1308 STAX D ENTER IT INTO TABLE
1309 INX D AND THE 2ND NAME
1310 INX H
1311 MOV A,M
1312 STAX D NAME NOW ENTERED
1313 INX D GET SET TO ENTER ADDRESS
1314 POP H RESTORE RTN ADDR
1315 XCHG .
1316 MOV M,E SET ADDR IN NOW
1317 INX H AND HI BYTE OF ADDR
1318 MOV M,D
1319 RET . NAME IS NOW ENTERED OR CLEARED
1320 *
1321 * -.-

```

```

**      PROGRAM DEVELOPMENT SYSTEM      **

SOLOS(TM) 77-03-27
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.
P.O. BOX 5260
SAN MATEO, CA 94402

PAGE 13

COPY SOLOS3/1

3 OF 3 *****

9999 *
1322 *
1323 *
1324 *
1325 *
1326 * THE FOLLOWING ROUTINES PROVIDE "BYTE BY BYTE" ACCESS
1327 * TO THE CASSETTE TAPES ON EITHER A READ OR WRITE BASIS.
1328 *
1329 * THE TAPE IS READ ONE BLOCK AT A TIME AND INDIVIDUAL
1330 * TRANSFERS OF DATA HANDLED BY MANAGING A BUFFER AREA.
1331 *
1332 * THE BUFFER AREA IS CONTROLLED BY A FILE CONTROL BLOCK
1333 * (FCB) WHOSE STRUCTURE IS:
1334 *
1335 *
1336 * 7 BYTES FOR EACH OF THE TWO FILES STRUCTURED AS
1337 * FOLLOWS:
1338 *
1339 * 1 BYTE - ACCESS CONTROL 00 IF CLOSED
1340 * FF IF READING
1341 * FE IF WRITING
1342 *
1343 * 1 BYTE - READ COUNTER
1344 * 1 BYTE - BUFFER POSITION POINTER
1345 * 2 BYTE - CONTROL HEADER ADDRESS
1346 * 2 BYTE - BUFFER LOCATION ADDRESS
1347 *
1348 *
1349 * THIS ROUTINE "OPENS" THE CASSETTE UNIT FOR ACCESS
1350 *
1351 * ON ENTRY: A - HAS THE TAPE UNIT NUMBER (1 OR 2)
1352 * HL - HAS USER SUPPLIED HEADER FOR TAPE FILE
1353 *
1354 *
1355 * NORMAL RETURN: ALL REGISTERS ARE ALTERED
1356 * BLOCK IS READY FOR ACCESS
1357 *
1358 * ERROR RETURN: CARRY BIT IS SET
1359 *
1360 * ERRORS: BLOCK ALREADY OPEN
1361 *
1362 *
1363 BOPEN PUSH H SAVE HEADER ADDRESS
1364 CALL LFCB GET ADDRESS OF FILE CONTROL
1365 JNZ TERE2 FILE WAS ALREADY OPEN
1366 MVI M,1 NOW IT IS
1367 INX H POINT TO READ COUNT
1368 MOV M,A ZERO
1369 INX H POINT TO BUFFER CURSOR
1370 MOV M,A PUT IN THE ZERO COUNT
1371 *
1372 * ALLOCATE THE BUFFER
1373 *

```

```

**      PROGRAM DEVELOPMENT SYSTEM      **

SOLOS(TM) 77-03-27
COPYRIGHT (C) 1977

C5ED 11 63 C8      1374      LXI      D,FBUF1  POINT TO BUFFER AREA
C5F0 3A 54 C8      1375      LDA      FNUMF  GET WHICH ONE WE ARE GOING TO USE
C5F3 82            1376      ADD      D
C5F4 57            1377      MOV      D,A    256 BIT ADD
C5F5 C1            1378 *
C5F6 B7            1379 UBUF  POP      B      HEADER ADDRESS
C5F7 C3 B6 C6      1380      ORA      A      CLEAR CARRY AND RETURN AFTER STORING PARAMS
C5F8 C3 B6 C6      1381      JMP      PSTOR  STORE THE VALUES
C5F9 C3 B6 C6      1382 *
C5FA E1            1383 *      GENERAL ERROR RETURN POINTS FOR STACK CONTROL
C5FB D1            1384 *
C5FC AF            1385 TERE2  POP      H
C5FD 37            1386 TERE1  POP      D
C5FE C9            1387 TERE0  XRA      A      CLEAR ALL FLAGS
C5FF C9            1388      STC      .      SET ERROR
C600 37            1389      RET
C601 D1            1390 *
C602 C9            1391 *
C603 CD 33 C6      1392 EOFER  DCR      A      SET MINUS FLAGS
C604 C9            1393      STC      .      AND CARRY
C605 C9            1394      POP      D      CLEAR THE STACK
C606 C8            1395      RET      .      THE FLAGS TELL ALL
C607 B7            1396 *
C608 3C            1397 *
C609 36 00         1398 *
C60A C8            1399 *
C60B C8            1400 *      THIS ROUTINE CLOSES THE FILE BUFFER TO ALLOW ACCESS
C60C 23            1401 *      FOR A DIFFERENT CASSETTE OR PROGRAM. IF THE FILE
C60D 23            1402 *      OPERATIONS WERE "WRITE" THEN THE LAST BLOCK IS WRITTEN
C60E 23            1403 *      OUT AND AN "END OF FILE" WRITTEN TO THE TAPE. IF
C60F 23            1404 *      THE OPERATIONS WERE "READS" THEN THE FILE IS JUST
C610 23            1405 *      MADE READY FOR NEW USE.
C611 23            1406 *
C612 23            1407 *      ON ENTRY: A - HAS WHICH UNIT (1 OR 2)
C613 23            1408 *
C614 23            1409 *      ERROR RETURNS: FILE WASN'T OPEN
C615 23            1410 *
C616 23            1411 *
C617 23            1412 PCLOS  CALL  LFCB  GET CONTROL BLOCK ADDRESS
C618 23            1413      RZ      .      WASN'T OPEN, CARRY IS SET FROM LFCB
C619 23            1414      ORA      A      CLEAR CARRY
C620 23            1415      INR      A      SET CONDITION FLAGS
C621 23            1416      MVI      M,0    CLOSE THE CONTROL BYTE
C622 23            1417      RZ      .      WE WERE READING...NOTHING MORE TO DO
C623 23            1418 *
C624 23            1419 *      THE FILE OPERATIONS WERE "WRITES"
C625 23            1420 *
C626 23            1421 *      PUT THE CURRENT BLOCK ON THE TAPE
C627 23            1422 *      (EVEN IF ONLY ONE BYTE!!)
C628 23            1423 *      THEN WRITE AN END OF FILE TO THE TAPE
C629 23            1424 *
C630 23            1425 *
C631 23            1426      INX      H

```

```

**      PROGRAM DEVELOPMENT SYSTEM      **

SOLOS(TM) 77-03-27
COPYRIGHT (C) 1977

C60D 23            1427      INX      H
C60E 7E            1428      MOV      A,M    GET CURSOR POSITION
C60F 7E            1429 *
C610 CD BF C6      1430      CALL  PLOAD  BC GET HEADER ADDRESS, DE BUFFER ADDRESS
C611 C5            1431      PUSH  B      HEADER TO STACK
C612 21 07 00      1432      LXI      H,BLKOF  OFFSET TO BLOCK SIZE
C613 09            1433      DAD      B
C614 B7            1434      ORA      A      TEST COUNT
C615 B7            1435      JZ      EOFW   NO BYTES...JUST WRITE EOF
C616 CA 2B C6      1436 *
C617 CA 2B C6      1437 *      WRITE LAST BLOCK
C618 CA 2B C6      1438 *
C619 CA 2B C6      1439      PUSH  H      SAVE BLOCK SIZE POINTER FOR EOF
C620 E5            1440      MOV      M,A    PUT IN COUNT
C621 77            1441      INX      H
C622 36 00         1442      MVI      M,0    ZERO THE HIGHER BYTE
C623 23            1443      INX      H
C624 73            1444      MOV      M,E    BUFFER ADDRESS
C625 23            1445      INX      H
C626 72            1446      MOV      M,D
C627 69            1447      MOV      H,B
C628 7C C7         1448      MOV      L,C    PUT HEADER ADDRESS IN HL
C629 E1            1449      CALL  WFBK  GO WRITE IT OUT
C630 E1            1450      POP      H    BLOCK SIZE POINTER
C631 E1            1451 *
C632 E1            1452 *      NOW WRITE END OF FILE TO CASSETTE
C633 E1            1453 *
C634 AF            1454 EOFW  XRA      A      PUT IN ZEROS FOR SIZE: EOF MARK IS ZERO BYTES
C635 77            1455      MOV      M,A
C636 23            1456      INX      H
C637 77            1457      MOV      M,A
C638 E1            1458      POP      H      HEADER ADDRESS
C639 C3 7C C7      1459      JMP      WFBK  WRITE IT OUT AND RETURN
C640 C3 7C C7      1460 *
C641 C3 7C C7      1461 *
C642 C3 7C C7      1462 *
C643 C3 7C C7      1463 *
C644 C3 7C C7      1464 *      THIS ROUTINE LOCATES THE FILE CONTROL BLOCK POINTED TO
C645 C3 7C C7      1465 *      BY REGISTER "A". ON RETURN HL POINT TO THE CONTROL BYT
C646 C3 7C C7      1466 *      AND REGISTER "A" HAS THE CONTROL WORD WITH THE FLAGS
C647 C3 7C C7      1467 *      SET FOR IMMEDIATE CONDITION DECISIONS.
C648 C3 7C C7      1468 *
C649 C3 7C C7      1469 *
C650 C3 7C C7      1470 LFCB  LXI      H,FCBAS  POINT TO THE BASE OF IT
C651 C3 7C C7      1471      RAR      .      MOVE THE 1 & 2 TO 0 & 1 LIKE COMPUTERS LIKE
C652 C3 7C C7      1472      ANI      1      SMALL NUMBERS ARE THE RULE
C653 C3 7C C7      1473      STA      FNUMF  CURRENT ACCESS FILE NUMBER
C654 C3 7C C7      1474      JZ      LFCE1  UNIT ONE (VALUE OF ZERO)
C655 C3 7C C7      1475      LXI      H,FCBA2  UNIT TWO--PT TO ITS FCB
C656 C3 7C C7      1476 LFCB1  EQU      $      HL PT TO PROPER FCB
C657 C3 7C C7      1477      MOV      A,M    PICK UP FLAGS FM FCB
C658 C3 7C C7      1478      ORA      A      SET FLAGS BASED ON CONTROL WORD
C659 C3 7C C7      1479      STC      .      SET CARRY IN CASE OF IMMEDIATE ERROR RETURN

```

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

C645 C9

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

```
1480 RET
1481 *
1482 *
1483 *
1484 *
1485 * READ TAPE BYTE ROUTINE
1486 *
1487 * ENTRY: - A - HAS FILE NUMBER
1488 * EXIT: NORMAL - A - HAS BYTE
1489 * ERROR
1490 * CARRY SET - IF FILE NOT OPEN OR
1491 * PREVIOUS OPERATIONS WERE WRITE
1492 * CARRY & MINUS - END OF FILE ENCOUNTERED
1493 *
1494 *
1495 *
1496 *
1497 RTBYT CALL LFCB LOCATE THE FILE CONTROL BLOCK
1498 RZ . FILE NOT OPEN
1499 INR A TEST IF FF
1500 JM TERE0 ERROR WAS WRITING
1501 MVI M,-1 SET IT AS READ (IN CASE IT WAS JUST OPENED)
1502 INX H
1503 MOV A,M GET READ COUNT
1504 PUSH H SAVE COUNT ADDRESS
1505 INX H
1506 CALL PLOAD GET THE OTHER PARAMETERS
1507 POP H
1508 ORA A
1509 JNZ GTBYT IF NOT EMPTY GO GET BYTE
1510 *
1511 * CURSOR POSITION WAS ZERO...READ A NEW BLOCK INTO
1512 * THE BUFFER.
1513 *
1514 RDNBLK PUSH D BUFFER POINTER
1515 PUSH H TABLE ADDRESS
1516 INX H
1517 CALL PHEAD PREPARE THE HEADER FOR READ
1518 CALL RFBK READ IN THE BLOCK
1519 JC TERE2 ERROR POP OFF STACK BEFORE RETURN
1520 POP H
1521 MOV A,E LOW BYTE OF COUNT (WILL BE ZERO IF 256)
1522 ORA D SEE IF BOTH ARE ZERO
1523 JZ EOFER BYTE COUNT WAS ZERO....END OF FILE
1524 MOV M,E NEW COUNT ( ZERO IS 256 AT THIS POINT)
1525 INX H BUFFER LOCATION POINTER
1526 MVI M,0
1527 DCX H
1528 MOV A,E COUNT TO A
1529 POP D GET BACK BUFFER ADDRESS
1530 *
1531 *
1532 *
```

C65C D5  
C65D E5  
C65E 23  
C65F CD A6 C6  
C662 CD C8 C6  
C665 DA FA C5  
C668 E1  
C669 7B  
C66A B2  
C66B CA FF C5  
C66E 73  
C66F 23  
C670 36 00  
C672 2B  
C673 7B  
C674 D1

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 15

```
1533 * THIS ROUTINE GETS ONE BYTE FROM THE BUFFER
1534 * AND RETURNS IT IN REGISTER "A". IF THE END
1535 * OF THE BUFFER IS REACHED IT MOVES THE POINTER
1536 * TO THE BEGINNING OF THE BUFFER FOR THE NEXT
1537 * LOAD.
1538 *
1539 GTBYT DCR A BUMP THE COUNT
1540 MOV M,A RESTORE IT
1541 INX H
1542 MOV A,M GET BUFFER POSITION
1543 INR M BUMP IT
1544 *
1545 ADD E
1546 MOV E,A DE NOW POINT TO CORRECT BUFFER POSITION
1547 JNC RT1
1548 INR D
1549 RT1 LDAX D GET CHARACTER FROM BUFFER
1550 ORA A CLEAR CARRY
1551 RET . ALL DONE
1552 *
1553 *
1554 *
1555 *
1556 * THIS ROUTINE IS USED TO WRITE A BYTE TO THE FILE
1557 *
1558 * ON ENTRY: A - HAS FILE NUMBER
1559 * B - HAS DATA BYTE
1560 *
1561 *
1562 WTBYT CALL LFCB GET CONTROL BLOCK
1563 RZ . FILE WASN'T OPEN
1564 INR A
1565 RZ . FILE WAS READ
1566 MVI M,OFEH SET IT TO WRITE
1567 INX H
1568 INX H
1569 MOV A,B GET CHARACTER
1570 PUSH PSW
1571 PUSH H SAVE CONTROL ADDRESS+2
1572 *
1573 * NOW DO THE WRITE
1574 *
1575 CALL PLOAD BC GETS HEADER ADDR, DE BUFFER ADDRESS
1576 POP H
1577 MOV A,M COUNT BYTE
1578 ADD E
1579 MOV E,A
1580 JNC WT1
1581 INR D
1582 WT1 POP PSW CHARACTER
1583 STAX D PUT CHR IN BUFFER
1584 ORA A CLEAR FLAGS
1585 INR M INCREMENT THE COUNT
```

C675 3D  
C676 77  
C677 23  
C678 7E  
C679 34  
  
C67A 83  
C67B 5F  
C67C D2 80 C6  
C67F 14  
C680 1A  
C681 B7  
C682 C9

C683 CD 33 C6  
C686 C8  
C687 3C  
C688 C8  
C689 36 FE  
C68B 23  
C68C 23  
C68D 78  
C68E F5  
C68F E5

C690 CD BF C6  
C693 E1  
C694 7E  
C695 83  
C696 5F  
C697 D2 9B C6  
C69A 14  
C69B F1  
C69C 12  
C69D B7  
C69E 34

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

```

C69F C0      1586 *      RNZ      RETURN IF COUNT DIDN'T ROLL OVER
              1587 *
              1588 *      THE BUFFER IS FULL. WRITE IT TO TAPE AND RESET
              1589 *      CONTROL BLOCK.
              1590 *
C6A0 CD A6 C6 1591 *      CALL PHEAD PREPARE THE HEADER
C6A3 C3 7C C7 1592 *      JMP WFBK WRITE IT OUT AND RETURN
              1593 *
              1594 *
              1595 *
              1596 *
              1597 *      THIS ROUTINE PUTS THE BLOCK SIZE (256) AND BUFFER
              1598 *      ADDRESS IN THE FILE HEADER.
              1599 *
C6A6 CD BF C6 1600 *      PHEAD CALL PLOAD GET HEADER AND BUFFER ADDRESSES
C6A9 C5        1601 *      PUSH B HEADER ADDRESS
C6AA 21 06 00 1602 *      LXI H,BLKOF-1 PSTOR DOES AN INCREMENT
C6AD 09        1603 *      DAD B HL POINT TO BLOCKSIZE ENTRY
C6AE 01 00 01 1604 *      LXI B,256
C6B1 CD B6 C6 1605 *      CALL PSTOR
C6B4 E1        1606 *      POP H HL RETURN WITH HEADER ADDRESS
C6B5 C9        1607 *      RET
              1608 *
              1609 *
C6B6 23        1610 *      PSTOR INX H
C6B7 71        1611 *      MOV M,C
C6B8 23        1612 *      INX H
C6B9 70        1613 *      MOV M,B
C6BA 23        1614 *      INX H
C6BB 73        1615 *      MOV M,E
C6BC 23        1616 *      INX H
C6BD 72        1617 *      MOV M,D
C6BE C9        1618 *      RET
              1619 *
              1620 *
C6BF 23        1621 *      PLOAD INX H
C6C0 4E        1622 *      MOV C,M
C6C1 23        1623 *      INX H
C6C2 46        1624 *      MOV B,M
C6C3 23        1625 *      INX H
C6C4 5E        1626 *      MOV E,M
C6C5 23        1627 *      INX H
C6C6 56        1628 *      MOV D,M
C6C7 C9        1629 *      RET
              1630 *
              1631 *
              1632 *
              1633 *
              1634 *
C6C8 CD DE C7 1635 *      THIS ROUTINE SETS THE CORRECT UNIT FOR SYSTEM READS
              1636 *      RFBLK CALL GTUNT SET UP A=UNIT WITH SPEED
              1637 *
              1638 *

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 16

```

              1639 *
              1640 *
              1641 *      TAPE READ ROUTINES
              1642 *
              1643 *      ON ENTRY: A HAS UNIT AND SPEED
              1644 *      HL POINT TO HEADER BLOCK
              1645 *      DE HAVE OPTIONAL PUT ADDRESS
              1646 *
              1647 *      ON EXIT: CARRY IS SET IF ERROR OCCURED
              1648 *      TAPE UNITS ARE OFF
              1649 *
              1650 *
C6CB D5        1651 *      RTAPE PUSH D SAVE OPTIONAL ADDRESS
C6CC 06 03     1652 *      MVI B,3 SHORT DELAY
C6CE CD EF C7 1653 *      CALL TON
C6D1 DB FB     1654 *      IN TDATA CLEAR THE UART FLAGS
              1655 *
C6D3 E5        1656 *      PTAP1 PUSH H HEADER ADDRESS
C6D4 CD 23 C7 1657 *      CALL RHEAD GO READ HEADER
C6D7 E1        1658 *      POP H
C6D8 DA 06 C7 1659 *      JC TERR IF AN ERROR OR ESC WAS RECEIVED
C6DB C2 D3 C6 1660 *      JNZ PTAP1 IF VALID HEADER NOT FOUND
              1661 *
              1662 *      FOUND A VALID HEADER NOW DO COMPARE
              1663 *
C6DE E5        1664 *      PUSH H GET BACK AND RESAVE ADDRESS
C6DF 11 1C C8 1665 *      LXI D,THEAD
C6E2 CD D2 C7 1666 *      CALL DHCMP COMPARE DE-HL HEADERS
C6E5 E1        1667 *      POP H
C6E6 C2 D3 C6 1668 *      JNZ PTAP1
              1669 *
              1670 *
C6E9 D1        1671 *      POP D OPTIONAL "PUT" ADDRESS
C6EA 7A        1672 *      MOV A,D
C6EB B3        1673 *      ORA E SEE IF DE IS ZERO
C6EC 2A 23 C8 1674 *      LHLD BLOCK GET BLOCK SIZE
C6EF EB        1675 *      XCHG ...TO DE
              1676 *      DE HAS HBLOCK...HL HAS USER OPTION
              1677 *      JNZ RTAP IF DE WAS ZERO GET TAPE LOAD ADDRESS
C6F0 C2 F6 C6 1678 *      LHLD LOADR GET TAPE LOAD ADDRESS
C6F3 2A 25 C8 1679 *
              1680 *
              1681 *      THIS ROUTINE READS "DE" BYTES FROM THE TAPE
              1682 *      TO ADDRESS HL. THE BYTES MUST BE FROM ONE
              1683 *      CONTIGUOUS PHYSICAL BLOCK ON THE TAPE.
              1684 *
              1685 *      HL HAS "PUT" ADDRESS
              1686 *      DE HAS SIZE OF TAPE BLOCK
              1687 *
C6F6 D5        1688 *      RTAP PUSH D SAVE SIZE FOR RETURN TO CALLING PROGRAM
              1689 *
C6F7          1690 *      EQU $ HERE TO LOOP RDING BLKS
C6F7 CD 15 C7 1691 *      CALL DCRCT DROP COUNT, B=LEN THIS BLK

```



\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

```

C6FA CA 10 C7 1692 JZ RTOFF ZERO=ALL DONE
1693 *
C6FD CD 44 C7 1694 CALL RHED1 READ THAT MANY BYTES
C700 DA 06 C7 1695 JC TERR IF ERROR OR ESC
C703 CA F7 C6 1696 JZ RTAP2 RD OK--READ SOME MORE
1697 *
1698 * ERROR RETURN
1699 *
C706 AF 1700 TERR XRA A
C707 37 1701 STC SET ERROR FLAGS
C708 C3 11 C7 1702 JMP RTOF1
1703 *
1704 *
C70B 06 01 1705 TOFF MVI B,1
C70D CD F1 C7 1706 CALL DELAY
C710 AF 1707 RTOFF XRA A
C711 D3 FA 1708 RTOF1 OUT TAPPT
C713 D1 1709 POP D RETURN BYTE COUNT
C714 C9 1710 RET
1711 *
1712 *
C715 1713 DCRCT EQU $ COMMON RTN TO COUNT DOWN BLK LENGTHS
C715 AF 1714 XRA A CLR FOR LATER TESTS
C716 47 1715 MOV B,A SET THIS BLK LEN=256
C717 B2 1716 ORA D IS AMNT LEFT < 256
C718 C2 20 C7 1717 JNZ DCRCT2 NO--REDUCE AMNT BY 256
C71B B3 1718 ORA E IS ENTIRE COUNT ZERO
C71C C8 1719 RZ ALL DONE--ZERO=THIS CONDITION
C71D 43 1720 MOV B,E SET THIS BLK LEN TO AMNT REMAINING
C71E 5A 1721 MOV E,D MAKE ENTIRE COUNT ZERO NOW
C71F C9 1722 RET ALL DONE (NON-ZERO FLAG)
C720 15 1723 DCRCT2 EQU $ REDUCE COUNT BY 256
C721 B7 1724 DCR D DROP BY 256
C722 C9 1725 ORA A FORCE NON-ZERO FLAG
1726 RET NON-ZERO=NOT DONE YET (BLK LEN=256)
1727 *
1728 *
1729 * READ THE HEADER
1730 *
C723 06 0A 1731 RHEAD MVI B,10 FIND 10 NULLS
C725 CD 5D C7 1732 RHEA1 CALL STAT
C728 D8 1733 RC IF ESCAPE
C729 DB FB 1734 IN TDATA IGNORE ERROR CONDITIONS
C72B B7 1735 ORA A ZERO?
C72C C2 23 C7 1736 JNZ RHEAD
C72F 05 1737 DCR B
C730 C2 25 C7 1738 JNZ RHEA1 LOOP UNTIL 10 IN A ROW
1739 *
1740 * WAIT FOR THE START CHARACTER
1741 *
C733 CD 6F C7 1742 SOHL CALL TAPIN
C736 D8 1743 RC ERROR OR ESCAPE
C737 FE 01 1744 CPI 1 AT LEAST 10 NULLS IMMEDIATELY FOLLOWED BY AN 01

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 17

```

C739 DA 33 C7 1745 JC SOHL STILL A NULL, KEEP WAITING
C73C C2 23 C7 1746 JNZ RHEAD NON-ZERO, START SEQUENCE OVER AGAIN
1747 *
1748 * NOW GET THE HEADER
1749 *
C73F 21 1C C8 1750 LXI H,THEAD POINT TO BUFFER
C742 06 10 1751 MVI B,HLEN LENGTH TO READ
1752 *
C744 1753 RHED1 EQU $ RD A BLOCK INTO HL FOR B BYTES
C744 0E 00 1754 MVI C,0 INIT THE CRC
C746 1755 RHED2 EQU $ LOOP HERE
C746 CD 6F C7 1756 CALL TAPIN GET A BYTE
C749 D8 1757 RC
C74A 77 1758 MOV M,A STORE IT
C74B 23 1759 INX H INCREMENT ADDRESS
C74C CD A8 C7 1760 CALL DOCRC GO COMPUTE THE CRC
C74F 05 1761 DCR B WHOLE HEADER YET?
C750 C2 46 C7 1762 JNZ RHED2 DO ALL THE BYTES
1763 *
1764 * THIS ROUTINE GETS THE NEXT BYTE AND COMPARES IT
1765 * TO THE VALUE IN REGISTER C. THE FLAGS ARE SET ON
1766 * RETURN.
1767 *
C753 CD 6F C7 1768 CALL TAPIN GET CRC BYTE
C756 A9 1769 XRA C CLR CARRY AND SET ZERO IF MATCH, ELSE NON-ZERO
C757 C8 1770 RZ CRC WAS FINE
C758 3A 11 C8 1771 LDA IGNCR GET POSSIBLE OVERRIDE CRC ERROR FLAG
C75B 3C 1772 INR A FF=IGNORE CRC ERRORS, ELSE PROCESS CRC ERROR
C75C C9 1773 RET
1774 *
1775 * THIS ROUTINE GETS THE NEXT AVAILABLE BYTE FROM THE
1776 * TAPE. WHILE WAITING FOR THE BYTE THE KEYBOARD IS TESTED
1777 * FOR AN ESC COMMAND. IF RECEIVED THE TAPE LOAD IS
1778 * TERMINATED AND A RETURN TO THE COMMAND MODE IS MADE.
1779 *
C75D DB FA 1780 STAT IN TAPPT TAPE STATUS PORT
C75F E6 40 1781 ANI TDR
C761 C0 1782 RNZ
C762 CD 1F C0 1783 CALL SINP CHECK INPUT
C765 CA 5D C7 1784 JZ STAT NOTHING THERE YET
C768 E6 7F 1785 ANI 7FH CLR PARITY 1ST
C76A C2 5D C7 1786 JNZ STAT NOT A MODE (OR EVEN CTL-0)
C76D 37 1787 STC SET ERROR FLAG
C76E C9 1788 RET AND RETURN
1789 *
1790 *
1791 *
C76F CD 5D C7 1792 TAPIN CALL STAT WAIT UNTIL A CHARACTER IS AVAILABLE
C772 D8 1793 RC
1794 *
C773 DB FA 1795 TREDY IN TAPPT TAPE STATUS
C775 E6 18 1796 ANI TFE+TOE DATA ERROR?
C777 DB FB 1797 IN TDATA GET THE DATA

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

```

C779 C8      1798      RZ      .      IF NO ERRORS
C77A 37      1799      STC      .      SET ERROR FLAG
C77B C9      1800      RET
1801 *
1802 *
1803 * THIS ROUTINE GETS THE CORRECT UNIT FOR SYSTEM WRITES
1804 * WFBLLK CALL GTUNT SET UP A WITH UNIT AND SPEED
1805 *
1806 *
1807 *
1808 * WRITE TAPE BLOCK ROUTINE
1809 *
1810 * ON ENTRY: A HAS UNIT AND SPEED
1811 * HL HAS POINTER TO HEADER
1812 *
1813 *
1814 * WTAPE EQU $ HERE TO WRITE TAPE
1815 * PUSH H SAVE HEADER ADDRESS
1816 * CALL WHEAD TURN ON, THEN WRITE HDR
1817 * POP H
1818 * LXI D,BLKOF OFFSET TO BLOCK SIZE IN HEADER
1819 * DAD D HL POINT TO BLOCK SIZE
1820 * MOV E,M
1821 * INX H
1822 * MOV D,M DE HAVE SIZE
1823 * INX H
1824 * MOV A,M
1825 * INX H
1826 * MOV H,M
1827 * MOV L,A HL HAVE STARTING ADDRESS
1828 *
1829 * THIS ROUTINE WRITES ONE PHYSICAL BLOCK ON THE
1830 * TAPE "DE" BYTES LONG FROM ADDRESS "HL".
1831 *
1832 *
1833 * WRLO1 EQU $ HERE FOR THE EXTRA PUSH
1834 * PUSH H A DUMMY PUSH FOR LATER EXIT
1835 * WTAPE2 EQU $ LOOP HERE UNTIL ENTIRE AMOUNT READ
1836 * CALL DCRCT DROP COUNT IN DE AND SET UP B W/LEN THIS BLK
1837 * JZ TOFF RETURNS ZERO IF ALL DONE
1838 * CALL WTBL WRITE BLOCK FOR BYTES IN B (256)
1839 * JMP WTAPE2 LOOP UNTIL ALL DONE
1840 *
1841 *
1842 * WRTAP PUSH PSW
1843 * WRWAT IN TAPPT TAPE STATUS
1844 * ANI TTBE IS TAPE READY FOR A CHAR YET
1845 * JZ WRWAT NO--WAIT
1846 * POP PSW YES--RESTORE CHAR TO OUTPUT
1847 * OUT TDATA SEND CHAR TO TAPE
1848 *
1849 * DOCRC EQU $ A COMMON CRC COMPUTATION ROUTINE
1850 * SUB C

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 18

```

C7A9 4F      1851      MOV      C,A
C7AA A9      1852      XRA      C
C7AB 2F      1853      CMA
C7AC 91      1854      SUB      C
C7AD 4F      1855      MOV      C,A
C7AE C9      1856      RET      .      ONE BYTE NOW WRITTEN
1857 *
1858 *
1859 * THIS ROUTINE WRITES THE HEADER POINTED TO BY
1860 * HL TO THE TAPE.
1861 *
1862 * WHEAD EQU $ HERE TO 1ST TURN ON THE TAPE
1863 * CALL WTON TURN IT ON, THEN WRITE HEADER
1864 * MVI D,50 WRITE 50 ZEROS
1865 * NULOP XRA A
1866 * CALL WRTAP
1867 * DCR D
1868 * JNZ NULOP
1869 *
1870 * MVI A,1
1871 * CALL WRTAP
1872 * MVI B,HLEN LENGTH TO WRITE OUT
1873 *
1874 * WTBL MVI C,0 RESET CRC BYTE
1875 * WLOOP MOV A,M GET CHARACTER
1876 * CALL WRTAP WRITE IT TO THE TAPE
1877 * DCR B
1878 * INX H
1879 * JNZ WLOOP
1880 * MOV A,C GET CRC
1881 * JMP WRTAP PUT IT ON THE TAPE AND RETURN
1882 *
1883 *
1884 * THIS ROUTINE COMPARES THE HEADER IN THEAD TO
1885 * THE USER SUPPLIED HEADER IN ADDRESS HL.
1886 * ON RETURN IF ZERO IS SET THE TWO NAMES COMPARED
1887 *
1888 * DHCMP MVI B,5
1889 * DHLOP LDAX D
1890 * CMP M
1891 * RNZ
1892 * DCR B
1893 * RZ . IF ALL FIVE COMPARED
1894 * INX H
1895 * INX D
1896 * JMP DHLOP
1897 *
1898 * GTUNT EQU $ SET A=SPEED + UNIT
1899 * LDA FNUMF GET UNIT
1900 * ORA A SEE WHICH UNIT
1901 * LDA TSPD BUT 1ST GET SPEED
1902 * JNZ GTUNT2 MAKE IT UNIT TWO
1903 * ADI TAPE2 THIS ONCE=UNIT 2, TWICE=UNIT 1

```

```

**          PROGRAM DEVELOPMENT SYSTEM          **

SOLOS(TM) 77-03-27
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.
P.O. BOX 5260
SAN MATEO, CA 94402

C7EA C6 40      1904 GTUN2 ADI TAPE2 UNIT AND SPEED NOW SET IN A
C7EC C9          1905 RET      ALL DONE
1906 *
C7ED 06 04      1907 WTON MVI B,4 SET LOOP DELAY (BIT LONGER ON A WRITE)
C7EF           1908 TON EQU $ HERE TO TURN A TAPE ON THEN DELAY
C7EF D3 FA      1909 OUT TAPPT GET TAPE MOVING, THEN DELAY
1910 *
C7F1 11 00 00   1911 DELAY LXI D,0
C7F4 1B         1912 DLOP1 DCX D
C7F5 7A         1913 MOV A,D
C7F6 B3         1914 ORA E
C7F7 C2 F4 C7   1915 JNZ DLOP1
C7FA 05         1916 DCR B
C7FB C2 F1 C7   1917 JNZ DELAY
C7FE C9         1918 RET
1919 *
1920 *
1921 ***** -- END OF PROGRAM--
1922 *
1923 *
1924 *
1925 *
1926 *      S Y S T E M   E Q U A T E S
1927 *
1928 *
1929 *      V D M   P A R A M E T E R S
1930 *
1931 VDMEM EQU OCC00H VDM SCREEN MEMORY
1932 *
1933 *
1934 *      K E Y B O A R D   S P E C I A L   K E Y   A S S I G N M E N T S
1935 *
1936 *      THESE DEFINITIONS ARE DESIGNED TO ALLOW
1937 *      COMPATABILITY WITH CUTER(TM). THESE ARE THE
1938 *      SAME KEYS WITH BIT 7 (X'80') STRIPPED OFF.
1939 *
009A 1940 DOWN EQU 9AH CTL-Z
0097 1941 UP EQU 97H CTL-W
0081 1942 LEFT EQU 81H CTL-A
0093 1943 RIGHT EQU 93H CTL-S
008B 1944 CLEAR EQU 8BH CTL-K
008E 1945 HOME EQU 8EH CTL-N
0080 1946 MODE EQU 80H CTL-@
005F 1947 BACKS EQU 5FH BACKSPACE
000A 1948 LF EQU 10
000D 1949 CR EQU 13
0020 1950 BLANK EQU
0020 1951 SPACE EQU BLANK
0018 1952 CX EQU 'X'-40H
001B 1953 ESC EQU 1BH
1954 *
1955 *      P O R T   A S S I G N M E N T S
1956 *

```

```

**          PROGRAM DEVELOPMENT SYSTEM          **

SOLOS(TM) 77-03-27
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.
P.O. BOX 5260
SAN MATEO, CA 94402

PAGE 19

00FA 1957 STAPT EQU OFAH STATUS PORT GENERAL
00F8 1958 SERST EQU OF8H SERIAL STATUS PORT
00F9 1959 SDATA EQU OF9H SERIAL DATA
00FD 1960 PDATA EQU OFDH PARALLEL DATA
00FC 1961 KDATA EQU OFCH KEYBOARD DATA
00FE 1962 DSTAT EQU OFEH VDM CONTROL PORT
00FA 1963 TAPPT EQU OFAH TAPE STATUS PORT
00FB 1964 TDATA EQU OFBH TAPE DATA PORT
00FF 1965 SENSE EQU OFFH SENSE SWITCHES
1966 *
1967 *
1968 *
1969 *      B I T   A S S I G N M E N T   M A S K S
1970 *
0001 1971 SCD EQU 1 SERIAL CARRIER DETECT
0002 1972 SDSR EQU 2 SERIAL DATA SET READY
0004 1973 SPE EQU 4 SERIAL PARITY ERROR
0008 1974 SFE EQU 8 SERIAL FRAMING ERROR
0010 1975 SOE EQU 16 SERIAL OVERRUN ERROR
0020 1976 SCTS EQU 32 SERIAL CLEAR TO SEND
0040 1977 SDR EQU 64 SERIAL DATA READY
0080 1978 STBE EQU 128 SERIAL TRANSMITTER BUFFER EMPTY
1979 *
0001 1980 KDR EQU 1 KEYBOARD DATA READY
0002 1981 PDR EQU 2 PARALLEL DATA READY
0004 1982 PXDR EQU 4 PARALLEL DEVICE READY
0008 1983 TFE EQU 8 TAPE FRAMING ERROR
0010 1984 TOE EQU 16 TAPE OVERFLOW ERROR
0040 1985 TDR EQU 64 TAPE DATA READY
0080 1986 TTBE EQU 128 TAPE TRANSMITTER BUFFER EMPTY
1987 *
0001 1988 SOK EQU 1 SCROLL OK FLAG
1989 *
0080 1990 TAPE1 EQU 80H 1=TURN TAPE ONE ON
0040 1991 TAPE2 EQU 40H 1=TURN TAPE TWO ON
1992 *
1993 *
1994 *
1995 *
1996 *      S Y S T E M   G L O B A L   A R E A
1997 *
C800 1998 ORG START+0800H RAM STARTS JUST AFTER ROM
1999 *
C800 2000 SYSRAM EQU $ START OF SYSTEM RAM
CBFF 2001 SYSTP EQU SYSRAM+3FFH STACK WORKS FM TOP DOWN
2002 *
2003 *
2004 *      P A R A M E T E R S   S T O R E D   I N   R A M
2005 *
C800 2006 UIPRT DS 2 USER DEFINED INPUT RTN IF NON ZERO
C802 2007 UOPRT DS 2 USER DEFINED OUTPUT RTN IF NON ZERO
C804 2008 DFLTS DS 2 DEFAULT PSUEDO I/O PORTS (ALWAYS ZERO IN SOLOS)
C806 2009 IPORT DS 1 CRNT INPUT PSUEDO PORT

```

PROGRAM DEVELOPMENT SYSTEM

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

```

C807 2010 OPORT DS 1 CRNT OUTPUT PSUEDO PORT
C808 2011 NCHAR DS 1 CURRENT CHARACTER POSITION
C809 2012 LINE DS 1 CURRENT LINE POSITION
C80A 2013 BOT DS 1 BEGINNING OF TEXT DISPLACEMENT
C80B 2014 SPEED DS 1 SPEED CONTROL BYTE
C80C 2015 ESCFL DS 1 ESCAPE FLAG CONTROL BYTE
C80D 2016 TSPD DS 1 CURRENT TAPE SPEED
C80E 2017 INPTR DS 2 FOR COMPATABILITY W/ CUTER
C810 2018 NUCNT DS 1 NUMBER OF NULLS AFTER CRLF
C811 2019 IGNCR DS 1 FF=IGNORE CRC ERRORS, ELSE NORMAL
2020 *
C812 2021 DS 10 ROOM FOR FUTURE EXPANSION
2022 *
2023 * *****
2024 * THIS IS THE HEADER LAYOUT
2025 * *****
2026 *
C81C 2027 THEAD DS 5 NAME
C821 2028 DS 1 THIS BYTE MUST BE ZERO
C822 2029 HTYPE DS 1 TYPE
C823 2030 BLOCK DS 2 BLOCK SIZE
C825 2031 LOADR DS 2 LOAD ADDRESS
C827 2032 XEQAD DS 2 AUTO EXECUTE ADDRESS
C829 2033 HSPR DS 3 SPARES
2034 *
0010 2035 HLEN EQU $-THEAD LENGTH OF HEADER
0007 2036 BLKOF EQU BLOCK-THEAD OFFSET TO BLOCK SIZE
C82C 2037 DHEAD DS HLEN A DUMMY HDR FOR COMPARES WHILE RD'ING
2038 *
2039 *
C83C 2040 CUTAB DS 6*4 ROOM FOR UP TO 6 CUSTOM USER COMMANDS
2041 *
2042 *
C854 2043 FNUMF DS 1 FOR CURRENT FILE OPERATIONS
C855 2044 FCBAS DS 7 1ST FILE CONTROL BLOCK
C85C 2045 FCBA2 DS 7 2ND FILE CONTROL BLOCK
C863 2046 FBUP1 DS 2*256 SYSTEM FILE BUFFER BASE
CAB4 2047 DS 81 THIS IS AN AREA USED BY CUTER
2048 USARE EQU $ START OF USER AREA
2049 * REMEMBER THAT THE STACK WORKS ITS WAY DOWN FROM
2050 * THE END OF THIS 1K RAM AREA.
2051 *
2052 * -*-

```

```

ADOUT C3E8 AINP C022 ALOAD C548 AOUT C01C
ARET C19B ARET1 C19D ARET2 C1A2 BACKS 005F
BLANK 0020 BLKOF 0007 BLOCK C823 BOPEN C5E0
BOT C80A BOUT C406 CHAR C094 CHPCK C05E
CHRLI C571 CLEAR 008B CLERA C1B4 CLIN1 C0FA
CLINE C0F4 COMN1 C1C0 COMND C1C9 COMTA C24A
CONT C1FF COPRC C205 CR 000D CREM C136
CRLF C2F9 CUR C0D1 CURET C1A6 CURSC C0CF
CUSE2 C5D3 CUSET C5BD CUTAB C83C CX 0018

```

PROGRAM DEVELOPMENT SYSTEM

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 20

```

DCRC2 C720 DCRCT C715 DEFLT C49A DELAY C7F1
DFLT C804 DHCMP C7D2 DHEAD C82C DHLOP C7D4
DISP C222 DISP1 C22B DISPD C599 DISPT C227
DLOOP C3C8 DLOP1 C7F4 DLP1 C3D3 DDCRC C7A8
DOWN 009A DSTAT 00FE DUMP C3BF ENLO1 C43C
ENLO3 C459 ENLOP C42B ENTER C423 Eofer C5FF
EOFW C62B ERAS1 C0DB ERAS3 C0EE ERR1 C480
ERR2 C481 ERRIT C2CB ERM C525 ERRO1 C2D6
ERRCT C2D2 ESC 001B ESCFL C80C ESCS C15F
ESCSP C168 EXEC C45E EXEC1 C461 FBUP1 C863
FCBA2 C85C FCBAS C855 FCLOS C00A FDCOM C231
FDCOU C22E FNUMF C854 FOPEN C007 GCLIN C1E4
GOBAC C06B GOBK C07C GTBYT C675 GTUN2 C7EA
GTUNT C7DE HBOUT C3ED HCONV C34D HCOV1 C35D
HEOU1 C414 HEOUT C40B HLEN 0010 HOME 008E
HSPR C829 HTYPE C822 IGNCR C811 INIT C001
INPTR C80E IOPRC C026 IPORT C806 ITAB C29A
KDATA 00FC KDR 0001 KSTAT C02E LEFT 0081
LF 000A LFCB C633 LFCB1 C642 LINE C809
LIST C539 LLIST C531 LOADR C825 MODE 0080
NAME C469 NAME1 C46E NAMES C466 NAOUT C550
NCHAR C808 NCOM C243 NEXT C080 NFIL C486
NLOOP C56A NUCNT C810 NULOP C7B4 NULOT C307
NCHAR C098 OK C0C1 OPORT C807 OTAB C292
OUTH C41F OUTPR C03B PASTA C2DD PBACK C13E
PCLOS C603 PCR C147 PCUR C10F PDATA 00FD
PDOWN C0CB PDR 0002 PERSE C0D5 PES C159
PHEAD C6A6 PHOME C0E5 PLEFT C10B PLF C14D
PLOAD C6BF PRIT C115 PROMP C2F1 PROUT C2E6
PSCAN C310 PSTOR C6B6 PTAP1 C6D3 PUP C104
PXDR 0004 RDBLK C013 RDBYT C00D RDNBL C65C
RETRN C004 RFBK C6C8 RHEA1 C725 RHEAD C723
RHED1 C744 RHED2 C746 RIGHT 0093 RT1 C680
RTAP C6F6 RTAP2 C6F7 RTAPE C6CB RTBYT C646
RTOF1 C711 RTOFF C710 SBLK C31B SBLK1 C31D
SCD 0001 SCHR C32E SCHR1 C330 SCONV C33A
SCROL C0AC SCTS 0020 SDATA 00F9 SDR 0040
SDROT C04A SDRS 0002 SECON C190 SENSE 00FF
SERST 00F8 SET C57A SETAB C2A2 SETCI C5A5
SETCO C5A9 SETCR C5B9 SETIN C59D SETNU C5B5
SETOT C5A1 SETSP C594 SETTY C5AD SETX C188
SETXQ C5B1 SETY C18C SFE 0008 SHE1 C343
SHEX C340 SINP C01F SOE 0010 SOHL C733
SOK 0001 SOUT C019 SPACE 0020 SPE 0004
SPEED C80B SROL C0B0 SSTAT C042 STAPT 00FA
START C000 STAT C75D STBE 0080 STRTA C1AF
STSPD C598 STUNT C4A2 SYSRA C800 SYSTP CBFF
TAERR C514 TAPE1 0080 TAPE2 0040 TAPIN C76F
TAPPT 00FA TASP D C58E TBL C273 TDATA 00FB
TDR 0040 TERE0 C5FC TERE1 C5FB TERE2 C5FA
TERM C367 TERM1 C373 TERM2 C3B9 TERR C706
TFE 0008 THEAD C81C TIMER C077 TIN C38B
TLIST C52B TLOA2 C4B5 TLOA3 C4C1 TLOAD C4A7

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

SOLOS(TM) 77-03-27  
COPYRIGHT (C) 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 21

TOE	0010	TOFF	C70B	TON	C7EF	TOUT	C388
TREDY	C773	TSAVE	C4E6	TSPD	C80D	TSRCH	C082
TTBE	0080	TXEQ	C4A6	UBUF	C5F5	UIPRT	C800
UOPRT	C802	UP	0097	USARE	CAB4	VDAD	C123
VDAD2	C120	VDADD	C11C	VDMEM	CC00	VDMOT	C054
WFBK	C77C	WHEAD	C7AF	WLOOP	C7C5	WRBLK	C016
WRBYT	C010	WRLO1	C790	WRTAP	C79D	WRWAT	C79E
WT1	C69B	WTAP2	C791	WTAPE	C77F	WTBL	C7C3
WTBYT	C683	WTLP1	C400	WTON	C7ED	XEQAD	C827

CONSOL<sup>TM</sup> Monitor Program Source Listing

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*  
 SOFTWARE TECHNOLOGY CORP.  
 P.O. BOX 5260  
 SAN MATEO, CA 94402  
 CONSOL (TM) 77-04-23  
 COPYRIGHT (C) 1976, 1977

PAGE 1

```

9999 COPY CONSOL1/1 1 OF 3 *****
0002 *
0003 *
0004 *
0005 *
0006 *
0007 *
0008 *
0009 *
0010 *
0011 *
0012 *
0013 *
0014 *
0015 *
0016 *
0017 *
0018 *
0019 *
0020 *
0021 *
0022 *
0023 *
0024 *
0025 *
0026 *
0027 *
0028 *
0029 *
0030 *
0031 *
0032 *
0033 *
0034 *
0035 *
0036 *
0037 *
0038 *
0039 *
0040 *
0041 *
0042 *
0043 *
0044 *
0045 *
0046 *
0047 *
0048 *
0049 *
0050 *
0051 *
  
```

CONSOL (TM)  
 COPYRIGHT (C) 1976, 1977  
 PROCESSOR TECHNOLOGY CORP.  
 ALL RIGHTS RESERVED!!!  
 \*\*\*\*\*  
 SYSTEM SOFTWARE  
 NOTE: CONSOL, SOLOS AND SOLED ARE REGISTERED  
 TRADEMARKS OF:  
 PROCESSOR TECHNOLOGY CORP.  
 EMERYVILLE, CALIF  
 ----- CONSOL -----  
 VERSION 1.3  
 RELEASE 77-04-23  
 THIS 2048 BYTE PROGRAM IS THE MINIMUM Sol STAND  
 ALONE OPERATING SYSTEM. IT IS CONFIGURED TO OPTIMIZE  
 THE CONVENIENCE AND POWER OF THE Sol-20 AND ONE OR TWO  
 CASSETTE RECORDERS IN STAND ALONE COMPUTER APPLICATIONS.  
 AUTO-STARTUP CODE  
 0047 START DB 0 FOUR PHASE WONDER  
 0048 INIT JMP STRTA SYSTEM RESTART ENTRY POINT  
 0049 \*  
 0050 \*  
 0051 \*

ENTRY POINTS

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*  
 SOFTWARE TECHNOLOGY CORP.  
 P.O. BOX 5260  
 SAN MATEO, CA 94402  
 CONSOL (TM) 77-04-23  
 COPYRIGHT (C) 1976, 1977

PAGE 2

```

0052 *
0053 *
0054 *
0055 *
0056 *
0057 *
0058 *
0059 *
0060 *
0061 *
0062 *
0063 *
0064 *
0065 *
0066 *
0067 *
0068 *
0069 *
0070 *
0071 *
0072 *
0073 *
0074 *
0075 *
0076 *
0077 *
0078 *
0079 *
0080 *
0081 *
0082 *
0083 *
0084 *
0085 *
0086 *
0087 *
0088 *
0089 *
0090 *
0091 *
0092 *
0093 *
0094 *
0095 *
0096 *
0097 *
0098 *
0099 *
0100 *
0101 *
0102 *
0103 *
0104 *
  
```

THESE JUMP POINTS ARE PROVIDED TO ALLOW COMMON ENTRY  
 LOCATIONS FOR ALL VERSIONS OF SOLOS. THEY ARE USED  
 EXTENSIVELY BY Sol SYSTEM PROGRAMS AND IT IS RECOMMENDED  
 THAT USER ROUTINES ACCESS SOLOS THROUGH THESE POINTS.  
 THIS JUMP TABLE IS PRESENT IN CONSOL TO PROVIDE  
 A COMPATIBLE SYSTEM BETWEEN CONSOL///SOLOS///CUTER.  
 RETRN JMP COMND RETURN TO SYSTEM ENTRY POINT  
 FOPEN JMP COMND ---NOT SUPPORTED---  
 FCLOS JMP COMND ---NOT SUPPORTED---  
 RDBYT JMP COMND ---NOT SUPPORTED---  
 WRBYT JMP COMND ---NOT SUPPORTED---  
 RDBLK JMP COMND ---NOT SUPPORTED---  
 WRBLK JMP COMND ---NOT SUPPORTED---  
 SYSTEM I/O ENTRY POINTS  
 THESE ROUTINES PERFORM SYSTEM I/O  
 THERE ARE TWO ENTRY TYPES:  
 SINP/SOUT REG "A" WILL BE SET TO THE STANDARD  
 SYSTEM PSEUDO PORT.  
 AINP/AOUT REG "A" MUST BE SET BY THE USER AND WILL  
 SPECIFY THE DESIRED PSEUDO PORT.  
 THE FOLLOWING ARE THE PSEUDO PORTS:  
 PORT DESCRIPTION  
 ---  
 0 KEYBOARD WHEN INPUT, AND VDM WHEN OUTPUT  
 1 SERIAL  
 2 PARALLEL  
 3 USER DEFINED  
 SOUT LDA OPORT SOUT ENTRY POINT  
 AOUT JMP OUTPR AOUT ENTRY POINT  
 SINP LDA IPORT SINP ENTRY POINT  
 AINP EQU \$ AINP ENTRY POINT  
 \*\*\*\*\* END OF SYSTEM ENTRY POINTS \*\*\*\*\*  
 PUSH H THIS IS ACTUALLY AINP  
 LXI H,ITAB  
 THIS ROUTINE PROCESSES THE I/O REQUESTS BY DISPATCHING  
 TO THE DRIVER REQUESTED IN REGISTER "A". ON ENTRY HL  
 HAVE THE PROPER DISPATCH TABLE.  
 IOPRC ANI 3 KEEP REGISTER "A" TO FOUR VALUES  
 RLC COMPUTE ENTRY ADDRESS  
 ADD L  
 MOV L,A WE HAVE ADDRESS  
 JMP DISPT DISPATCH TO IT

C004 C3 69 C1  
 C007 C3 69 C1  
 C00A C3 69 C1  
 C00D C3 69 C1  
 C010 C3 69 C1  
 C013 C3 69 C1  
 C016 C3 69 C1  
 C019 3A 07 C8  
 C01C C3 3B C0  
 C01F 3A 06 C8  
 C022  
 C022 E5  
 C023 21 1B C2  
 C026 E6 03  
 C028 07  
 C029 85  
 C02A 6F  
 C02B C3 B8 C1



```

**      PROGRAM DEVELOPMENT SYSTEM      **
CONSOL (TM) 77-04-23
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.
P.O. BOX 5260
SAN MATEO, CA 94402

PAGE 3

0105 *
0106 *
0107 *
0108 *
0109 *
0110 *
0111 *
0112 *      ----- Sol SYSTEM I/O ROUTINES -----
0113 *      THIS ROUTINE IS A MODEL OF ALL INPUT ROUTINES WITHIN
0114 *      SOLOS. EACH ROUTINE FIRST TESTS THE STATUS INPUT FOR
0115 *      DATA AVAILABLE. IF NO CHARACTER HAS BEEN RECEIVED THE
0116 *      ROUTINE RETURNS WITH THE ZERO FLAG SET. OTHERWISE THE
0117 *      CHARACTER IS INPUT AND A RETURN MADE WITH THE CHARACTER
0118 *      IN THE ACCUMULATOR AND THE ZERO FLAG RESET.
0119 *
0120 *      KEYBOARD INPUT DRIVER
0121 *
0122 * KSTAT IN STAPT GET STATUS WORD
0123 * CMA . INVERT IT FOR PROPER RETURN
0124 * ANI KDR TEST KEYBOARD BIT
0125 * RZ . ZERO IF NO CHARACTER RECEIVED
0126 *
0127 * IN KDATA GET CHARACTER
0128 * RET . GO BACK WITH IT
0129 *
0130 *
0131 *      THIS JUMP IS PART OF THE AUTO START UP CODE
0132 *
0133 * DB 0 *****VERIFY ADDR=C037*****
0134 * JMP INIT
0135 *
0136 *
0137 *      JUMP TABLE OUTPUT ROUTINES
0138 *
0139 *      THIS ROUTINE SETS UP THE DISPATCH TABLE FOR OUTPUT
0140 *      ROUTINES. THE CHARACTER FOR OUTPUT IS IN REGISTER "R".
0141 *      OUTPUT IS MADE TO THE DRIVER POINTED TO BY THE REGISTER
0142 *      "A". THE DEVICE DRIVERS ARE DEFINED AS FOLLOWS:
0143 *
0144 *
0145 *      0 - DISPLAY SCREEN
0146 *      1 - SERIAL OUTPUT PORT
0147 *      2 - PARALLEL OUTPUT PORT
0148 *      3 - USER DEFINED OR ERROR FLAG
0149 *
0150 * ENTRY AT: SOUT SELECTS CURRENT OUTPUT DEVICE.
0151 *           AOUT SELECTS DEVICE IN REGISTER 'A'
0152 *
0153 * OUTPR PUSH H
0154 * LXI H,OTAB POINT TO OUTPUT TABLE
0155 * JMP IOPRC AND DISPATCH TO OUTPUT ROUTINE
0156 *
0157 *

```

C02E DB FA  
C030 2F  
C031 E6 01  
C033 C8

C034 DB FC  
C036 C9

C037 00  
C038 C3 01 C0

C03B E5  
C03C 21 13 C2  
C03F C3 26 C0

```

**      PROGRAM DEVELOPMENT SYSTEM      **
CONSOL (TM) 77-04-23
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.
P.O. BOX 5260
SAN MATEO, CA 94402

PAGE 4

0158 *
0159 *
0160 *
0161 *      SERIAL INPUT DRIVER
0162 * SSTAT IN SERST GET SERIAL STATUS WORD
0163 * ANI SDR TEST FOR SERIAL DATA READY
0164 * RZ . FLAGS ARE SET
0165 *
0166 * IN SDATA GET DATA BYTE
0167 * RET . WE HAVE IT
0168 *
0169 *
0170 *      SERIAL DATA OUTPUT
0171 *
0172 * SDROT IN SERST GET PORT STATUS
0173 * RAL . PUT HIGH BIT IN CARRY
0174 * JNC SDROT LOOP UNTIL TRANSMITTER BUFFER IS EMPTY
0175 * MOV A,B GET THE CHARACTER BACK
0176 * OUT SDATA SEND IT OUT
0177 * RET . AND WE'RE DONE
0178 *
0179 *
0180 *
0181 *
0182 *
0183 *      VIDEO DISPLAY ROUTINES
0184 *
0185 *
0186 *      THESE ROUTINES ALLOW FOR STANDARD VIDEO TERMINAL
0187 *      OPERATIONS. ON ENTRY, THE CHARACTER FOR OUTPUT IS IN
0188 *      REGISTER B AND ALL REGISTERS EXCEPT "A" AND FLAGS ARE
0189 *      UNALTERED ON RETURN.
0190 *
0191 * VDMOX EQU $ SPECIAL ENTRY POINT TO IGN CTL CHARS FM USER
0192 * MOV A,B GET THE CHAR
0193 * CPI LF IS IT A CTL CHAR TO BE IGNORED???
0194 * RC YES--IGNORE EM
0195 * CPI ESC ALSO THIS ONE TO BE IGNORED
0196 * RZ YUP, IGNORE IT
0197 *
0198 *
0199 * VDMOT PUSH H SAVE MOST REGISTERS
0200 * PUSH D
0201 * PUSH B
0202 *
0203 *
0204 * CHPK MOV A,B SAVE IN B...STRIP PARITY BEFORE SCREEN!
0205 * ANI 7FH CLR PARITY TO LOCATE IN TBL
0206 * MOV B,A KEEP IT W/OUT PARITY IN B TOO
0207 * JZ GOBK DO A QUICK EXIT IF A NULL
0208 * LXI H,TBL POINT TO SPECIAL CHARACTER TABLE
0209 * CALL TSRCH GO PROCESS
0210 *

```

C042 DB F8  
C044 E6 40  
C046 C8

C047 DB F9  
C049 C9

C04A DB F8  
C04C 17  
C04D D2 4A C0  
C050 78  
C051 D3 F9  
C053 C9

C054  
C054 78  
C055 FE 0A  
C057 D8  
C058 FE 1B  
C05A C8

C05B E5  
C05C D5  
C05D C5

C05E 78  
C05F E6 7F  
C061 47  
C062 CA 72 C0  
C065 21 F7 C1  
C068 CD 78 C0

PROGRAM DEVELOPMENT SYSTEM

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 5

```

C06B CD 12 C1 0211 GOBACK CALL VDADD GET SCREEN ADDRESS
C06E 7E 0212 MOV A,M GET PRESENT CURSOR CHARACTER
C06F F6 80 0213 ORI 80H
C071 77 0214 MOV M,A CURSOR IS BACK ON
C072 C1 0215 GOBK POP B
C073 D1 0216 POP D RESTORE REGISTERS
C074 E1 0217 POP H
C075 C9 0218 RET EXIT FROM VDMOT
0219 *
C076 23 0220 NEXT INX H
C077 23 0221 INX H
0222 *
0223 *
0224 * THIS ROUTINE SEARCHES THROUGH A SINGLE CHARACTER
0225 * TABLE FOR A MATCH TO THE CHARACTER IN "B". IF FOUND
0226 * A DISPATCH IS MADE TO THE ADDRESS FOLLOWING THE MATCHED
0227 * CHARACTER. IF NOT FOUND THE CHARACTER IS DISPLAYED ON
0228 * THE MONITOR.
0229 *
C078 7E 0230 TSRCH MOV A,M GET CHR FROM TABLE
C079 B7 0231 ORA A
C07A CA 8A C0 0232 JZ CHAR ZERO IS THE LAST
C07D B8 0233 CMP B TEST THE CHR
C07E 23 0234 INX H POINT FORWARD
C07F C2 76 C0 0235 JNZ NEXT
C082 E5 0236 PUSH H FOUND ONE...SAVE ADDRESS
C083 CD 2C C1 0237 CALL CREM REMOVE CURSOR
C086 E3 0238 XTHL GET DISPATCH ADDRESS TO HL
C087 C3 B8 C1 0239 JMP DISPT DISPATCH NOW
0240 *
0241 * PUT CHARACTER TO SCREEN
0242 *
C08A 78 0243 CHAR MOV A,B GET CHARACTER
C08B FE 7F 0244 CPI 7FH IS IT A DEL?
C08D C8 0245 RZ GO BACK IF SO
0246 *
0247 *
0248 *
C08E 0249 OCHAR EQU $ ACTUALLY PUT CHAR TO SCREEN NOW
C08E CD 12 C1 0250 CALL VDADD GET SCREEN ADDRESS
C091 70 0251 MOV M,B PUT CHR ON SCREEN
0252 *
C092 3A 08 C8 0253 LDA NCHAR GET CHARACTER POSITION
C095 FE 3F 0254 CPI 63 END OF LINE?
C097 DA B7 C0 0255 JC OK
C09A 3A 09 C8 0256 LDA LINE
C09D FE 0F 0257 CPI 15 END OF SCREEN?
C09F C2 B7 C0 0258 JNZ OK
0259 *
0260 * END OF SCREEN...ROLL UP ONE LINE
0261 *
C0A2 AF 0262 SCROLL XRA A
C0A3 32 08 C8 0263 STA NCHAR BACK TO FIRST CHAR POSITION

```

PROGRAM DEVELOPMENT SYSTEM

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 6

```

C0A6 4F 0264 SROL MOV C,A
C0A7 CD 19 C1 0265 CALL VDAD CALCULATE LINE TO BE BLANKED
C0AA AF 0266 XRA A
C0AB CD F0 C0 0267 CALL CLIN1 CLEAR IT
C0AE 3A 0A C8 0268 LDA BOT
C0B1 3C 0269 INR A
C0B2 E6 0F 0270 ANI 0FH
C0B4 C3 E4 C0 0271 JMP ERAS3
0272 *
0273 * INCREMENT LINE COUNTER IF NECESSARY
0274 *
C0B7 3A 08 C8 0275 OK LDA NCHAR GET CHR POSITION
C0BA 3C 0276 INR A
C0BB E6 3F 0277 ANI 3FH MOD 64 AND WRAP
C0BD 32 08 C8 0278 STA NCHAR
C0C0 C0 0279 RNZ
C0C1 3A 09 C8 0280 PDOWN EQU $ DIDN'T HIT END OF LINE, OK
C0C4 3C 0281 LDA LINE CURSOR DOWN ONE LINE HERE
C0C5 E6 0F 0282 INR A GET THE LINE COUNT
C0C7 32 09 C8 0283 CURSC ANI 0FH MOD 15 INCREMENT
C0CA C9 0284 CUR STA LINE STORE THE NEW
0285 RET
0286 *
0287 * ERASE SCREEN
0288 *
C0CB 21 00 CC 0289 PERSE LXI H,VDMM POINT TO SCREEN
C0CE 36 A0 0290 MVI M,80H+ THIS IS THE CURSOR
0291 *
C0D0 23 0292 INX H BUMP 1ST
C0D1 36 20 0293 ERAS1 EQU $ LOOPS HERE TO ERASE SCREEN
C0D3 23 0294 MVI M, BLANK IT OUT
C0D4 7C 0295 INX H NEXT
C0D5 FE D0 0296 MOV A,H SEE IF END OF SCREEN YET
C0D7 DA D1 C0 0297 CPI 0DOH ?
C0DA 37 0298 JC ERAS1 NO--KEEP BLANKING
0299 STC CARRY WILL SAY COMPLETE ERASE
0300 *
C0DB 3E 00 0301 PHOME MVI A,0 RESET CURSOR--CARRY=ERASE, ELSE HOME
C0DD 32 09 C8 0302 STA LINE ZERO LINE
C0E0 32 08 C8 0303 STA NCHAR LEFT SIDE OF SCREEN
C0E3 D0 0304 RNC IF NO CARRY, WE ARE DONE WITH HOME
0305 *
C0E4 D3 FE 0306 ERAS3 OUT DSTAT RESET SCROLL PARAMETERS
C0E6 32 0A C8 0307 STA BOT BEGINNING OF TEXT OFFSET
C0E9 C9 0308 RET
0309 *
0310 *
C0EA CD 12 C1 0311 CLINE CALL VDADD GET CURRENT SCREEN ADDRESS
C0ED 3A 08 C8 0312 LDA NCHAR CURRENT CURSOR POSITION
C0F0 FE 40 0313 CLIN1 CPI 64 NO MORE THAN 63
C0F2 D0 0314 RNC ALL DONE
C0F3 36 20 0315 MVI M, ALL SPACED OUT
C0F5 23 0316 INX H

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 7

```

COF6 3C      0317      INR      A
COF7 C3 F0 C0 0318      JMP      CLIN1 LOOP TO END OF LINE
               0319 *
               0320 *
               0321 * ROUTINE TO MOVE THE CURSOR UP ONE LINE
               0322 *
COFA 3A 09 C8 0323 PUP      LDA      LINE GET LINE COUNT
COFD 3D      0324      DCR      A
COFE C3 C5 C0 0325      JMP      CURSC MERGE TO HANDLE CURSOR
               0326 *
               0327 * MOVE CURSOR LEFT ONE POSITION
               0328 *
C101 3A 08 C8 0329 PLEFT     LDA      NCHAR
C104 3D      0330      DCR      A
               0331 PCUR      EQU      $ CURSOR ON SAME LINE
C105 E6 3F      0332      ANI      3FH LET CURSOR WRAP
C107 32 08 C8 0333      STA      NCHAR UPDATED CURSOR
C10A C9      0334      RET
               0335 *
               0336 * CURSOR RIGHT ONE POSITION
               0337 *
C10B 3A 08 C8 0338 PRIT      LDA      NCHAR
C10E 3C      0339      INR      A
C10F C3 05 C1 0340      JMP      PCUR
               0341 *
               0342 * ROUTINE TO CALCULATE SCREEN ADDRESS
               0343 *
               0344 * ENTRY AT: RETURNS:
               0345 *
               0346 * VDADD CURRENT SCREEN ADDRESS
               0347 * VDAD2 ADDRESS OF CURRENT LINE, CHAR 'C'
               0348 * VDAD LINE 'A', CHARACTER POSITION 'C'
               0349 *
C112 3A 08 C8 0350 VDADD     LDA      NCHAR GET CHARACTER POSITION
C115 4F      0351      MOV      C,A 'C' KEEPS IT
C116 3A 09 C8 0352 VDAD2     LDA      LINE LINE POSITION
C119 6F      0353 VDAD      MOV      L,A INTO 'L'
C11A 3A 0A C8 0354      LDA      BOT GET TEXT OFFSET
C11D 85      0355      ADD      L ADD IT TO THE LINE POSITION
C11E 0F      0356      RRC      . TIMES TWO
C11F 0F      0357      RRC      . MAKES FOUR
C120 6F      0358      MOV      L,A L HAS IT
C121 E6 03      0359      ANI      3 MOD THREE FOR LATER
C123 C6 CC      0360      ADI      <VDMEM LOW SCREEN OFFSET
C125 67      0361      MOV      H,A NOW H IS DONE
C126 7D      0362      MOV      A,L TWIST L'S ARM
C127 E6 C0      0363      ANI      0COH
C129 81      0364      ADD      C
C12A 6F      0365      MOV      L,A
C12B C9      0366      RET      . H & L ARE NOW PERVERTED
               0367 *
               0368 * ROUTINE TO REMOVE CURSOR
               0369 *

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 8

```

C12C CD 12 C1 0370 CREM      CALL      VDADD GET CURRENT SCREEN ADDRESS
C12F 7E      0371      MOV      A,M
C130 E6 7F      0372      ANI      7FH STRIP OFF THE CURSOR
C132 77      0373      MOV      M,A
C133 C9      0374      RET
               0375 *
               0376 * ROUTINE TO BACKSPACE
               0377 *
C134 CD 01 C1 0378 PBACK     CALL      PLEFT
C137 CD 12 C1 0379      CALL      VDADD GET SCREEN ADDRESS
C13A 36 20      0380      MVI      M, ' ' PUT A BLANK THERE
C13C C9      0381      RET
               0382 *
               0383 * ROUTINE TO PROCESS A CARRIAGE RETURN
               0384 *
C13D CD EA C0 0385 PCR      CALL      CLINE CLEAR FROM CURRENT CURSOR TO END OF LINE
               0386 * NOTE THAT "A" COMES BACK-64 AND IS CLEARED BY PCUR
C140 C3 05 C1 0387      JMP      PCUR AND STORE THE NEW VALUE
               0388 *
               0389 * ROUTINE TO PROCESS LINEFEED
               0390 *
C143 3A 09 C8 0391 PLF      LDA      LINE GET LINE COUNT
C146 3C      0392      INR      A
C147 E6 0F      0393      ANI      15 SEE IF IT WRAPPED AROUND
C149 C2 C7 C0 0394      JNZ      CUR NO--NO NEED TO SCROLL
C14C C3 A6 C0 0395      JMP      SROL
               0396 *
               0397 *
               0398 *
               0399 * START UP SYSTEM
               0400 *
               0401 * CLEAR SCREEN AND THE FIRST 256 BYTES OF GLOBAL RAM
               0402 * THEN ENTER THE COMMAND MODE.
C14F AF      0403 STRTA     XRA      A
C150 4F      0404      MOV      C,A
C151 21 00 C8 0405      LXI      H,SYSRAM CLEAR THE FIRST PAGE
               0406 *
C154 77      0407 CLERA     MOV      M,A
C155 23      0408      INX      H
C156 0C      0409      INR      C
C157 C2 54 C1 0410      JNZ      CLERA
               0411 *
C15A 31 FF CB 0412      LXI      SP,SYSTP SET UP THE STACK FOR CALL
C15D CD CB C0 0413      CALL     PERSE
               C160
C160 AF      0414 ERROT     EQU      $ INVALID I/O PORTS COME HERE FOR CONSOL
C161 D3 FA      0415 COMN1    XRA      A
C163 32 07 C8 0416      OUT      STAPT BE SURE TAPES ARE OFF
C166 32 06 C8 0417      STA      OPORT
               0418      STA      IPORT
               0419 *
               0420 *
               0421 *
               0422 * --- COMMAND MODE ---

```

PROGRAM DEVELOPMENT SYSTEM

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 9

```

0423 *
0424 *
0425 * THIS ROUTINE GETS AND PROCESSES COMMANDS
0426 *
0427 COMND LXI SP,SYSTP SET STACK POINTER
0428 CALL PROMPT PUT PROMPT ON SCREEN
0429 CALL GCLIN GET COMMAND LINE
0430 CALL COPRC PROCESS THE LINE
0431 JMP COMND OVER AND OVER
0432 *
0433 *
0434 *
0435 * THIS ROUTINE READS A COMMAND LINE FROM THE SYSTEM
0436 * KEYBOARD
0437 *
0438 * C/R TERMINATES THE SEQUENCE ERASING ALL CHARS TO THE
0439 * RIGHT OF THE CURSOR
0440 * L/F TERMINATES THE SEQUENCE
0441 * MODE RESTARTS THE COMMAND LINE.
0442 *
0443 GCLIN CALL KSTAT GET A CHAR FM SOL KEYBOARD
0444 JZ GCLIN
0445 ANI 7FH CLEAR PARITY BIT
0446 JZ COMN1 THIS WAS A MODE (OR EVEN CTL-8)
0447 MOV B,A
0448 CPI CR CARRIAGE RETURN
0449 JZ CLINE YES--DONE WITH LINE
0450 CPI LF LINE FEED
0451 RZ YES--DONE WITH LINE, LEAVE AS IS
0452 CPI 7FH DELETE CHR?
0453 JNZ CONT
0454 MVI B,BACKS REPLACE IT
0455 *
0456 CONT CALL VDMOT OUTPUT TO VDM ALWAYS
0457 JMP GCLIN
0458 *
0459 *
0460 * FIND AND PROCESS COMMAND
0461 *
0462 COPRC CALL CREM REMOVE THE CURSOR
0463 MVI C,1 SET FOR CHARACTER POSITION
0464 CALL VDAD2 GET SCREEN ADDRESS
0465 XCHG
0466 LXI H,START MAKE SURE HL PT TO SOLOS START
0467 PUSH H SAVE IT FOR LATER DISPT
0468 CALL SCHR SCAN PAST BLANKS
0469 JZ ERR1 NO COMMAND?
0470 XCHG HL HAS FIRST CHR
0471 *
0472 LXI D,COMTAB POINT TO COMMAND TABLE
0473 CALL FDCOM SEE IF IN PRIMARY COMMAND TABLE
0474 JZ ERR2 NOT VALID, ERROR
0475 INX D BUMP TO PTR OF RTN

```

PROGRAM DEVELOPMENT SYSTEM

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 10

```

C1B7 EB 0476 XCHG HL PT TO RTN ADDR
0477 *
0478 *
0479 * THIS IS THE DISPATCH ROUTINE.
0480 * HL PT TO RTN ADDRESS, HL WILL BE RESTORED FM STACK
0481 * SO THAT HL ARE RESTORED BEFORE DISPATCH.
0482 *
0483 DISPT EQU $ OFF TO A ROUTINE
0484 MOV A,M LO ADDR
0485 INX H
0486 MOV H,M HI ADDR
0487 MOV L,A HL NOW COMPLETE
0488 DISP1 EQU $ HERE TO GO OFF TO HL
0489 XTHL XCHG HL W/HL ON STACK
0490 MOV A,L ALSO COPY HERE FOR SETS
0491 RET AND GO OFF TO THE RTN
0492 *
0493 *
0494 GOBAS LXI H,0 THIS EXECUTES SOL BASIC LOCATION ZERO
0495 JMP EXEC1 BECAUSE HL MUST BE PASSED PROPERLY
0496 *
0497 * THIS ROUTINE SEARCHES THROUGH A TABLE, POINTED TO
0498 * BY 'DE', FOR A DOUBLE CHARACTER MATCH OF THE 'HL'
0499 * MEMORY CONTENT. IF NO MATCH IS FOUND THE SCAN ENDS
0500 * WITH HL POINTING TO ORIGINAL VALUE AND ZERO FLAG SET.
0501 *
0502 *
0503 FDCOM LDAX D
0504 ORA A TEST FOR TABLE END
0505 RZ NOT FOUND..COMMAND ERROR
0506 PUSH H SAVE START OF SCAN ADDRESS
0507 CMP M TEST FIRST CHR
0508 INX D
0509 JNZ NCOM
0510 *
0511 INX H
0512 LDAX D
0513 CMP M NOW SECOND CHARACTER
0514 JNZ NCOM GOODNESS
0515 *
0516 POP H RESTORE ORIGINAL SCAN ADDR
0517 ORA A SET NON-ZERO FLAG SAYING FOUND
0518 RET WITH NON-ZERO SET
0519 *
0520 *
0521 NCOM INX D GO TO NEXT ENTRY
0522 INX D
0523 INX D
0524 POP H GET BACK ORIGINAL ADDRESS
0525 JMP FDCOM CONTINUE SEARCH
0526 *
0527 *
0528 * COMMAND TABLE

```

PROGRAM DEVELOPMENT SYSTEM

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 11

```

0529 *
0530 * THIS TABLE DESCRIBES THE VALID COMMANDS FOR SOLOS
0531 *
0532 COMTAB ASC 'TE' TERMINAL MODE
0533 DW TERM
0534 ASC 'DU' DUMP
0535 DW DUMP
0536 ASC 'EN' ENTR
0537 DW ENTER
0538 ASC 'EX' EXEC
0539 DW EXEC
0540 ASC 'GE' GET A FILE
0541 DW TLOAD
0542 ASC 'BA' BASIC
0543 DW GOBAS
0544 DB 0 END OF TABLE MARK
0545 *
0546 *
0547 *
0548 *
0549 * THIS TABLE DEFINES THE CHARACTERS FOR SPECIAL
0550 * PROCESSING. IF THE CHARACTER IS NOT IN THE TABLE IT
0551 * GOES TO THE SCREEN.
0552 *
0553 TBL DB CLEAR-80H SCREEN
0554 DW PERSE
0555 DB UP-80H CURSOR
0556 DW PUP
0557 DB DOWN-80H
0558 DW PDOWN
0559 DB LEFT-80H
0560 DW PLEFT
0561 DB RIGHT-80H
0562 DW PRIT
0563 DB HOME-80H
0564 DW PHOME
0565 DB CR CARRIAGE RETURN
0566 DW PCR
0567 DB LF LINE FEED
0568 DW PLF
0569 DB BACKS BACK SPACE
0570 DW PBACK
0571 DB 0 END OF TABLE
0572 *
0573 *
0574 *
0575 *
0576 OTAB DW VDMOX SPECIAL VDM ENTRY POINT TO IGN CTL CHARS
0577 DW SDROT SERIAL OUTPUT
0578 DW ERROT PARALLEL NOT SUPPORTED BY CONSOL
0579 DW ERROT USER ROUTINE NOT SUPPORTED BY CONSOL
0580 *
0581 *

```

OUTPUT DEVICE TABLE

PROGRAM DEVELOPMENT SYSTEM

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 12

```

0582 * INPUT DEVICE TABLE
0583 *
0584 ITAB DW KSTAT KEYBOARD INPUT
0585 DW SSTAT SERIAL INPUT
0586 DW ERROT PARALLEL NOT SUPPORTED BY CONSOL
0587 DW ERROT USER ROUTINE NOT SUPPORTED BY CONSOL
0588 *
0589 *
0590 *
0591 * *-
0592 * COPY CONSOL2/1
0593 *
0594 *
0595 * OUTPUT A CRLF FOLLOWED BY A PROMPT
0596 PROMPT CALL CRLF
0597 MVI B,'>' THE PROMPT
0598 JMP SOUT PUT IT ON THE SCREEN
0599 *
0600 *
0601 CRLF MVI B,LF LINE FEED
0602 CALL VDMOT
0603 MVI B,CR CARRIAGE RETURN
0604 JMP VDMOT
0605 *
0606 *
0607 * SCAN OFF OPTIONAL PARAMETER. IF PRESENT RETURN WITH
0608 * VALUE IN HL AND COPY OF "L" IN "A". IF NOT PRESENT
0609 * RETURN WITH A "1" IN "A" AND HL UNTOUCHED.
0610 *
0611 PSCAN CALL SBLK
0612 RZ IF NONE
0613 CALL SHEX CONVERT VALUE
0614 RET
0615 *
0616 *
0617 * SCAN OVER UP TO 12 CHARACTERS LOOKING FOR A BLANK
0618 *
0619 SBLK MVI C,12 MAXIMUM COMMAND STRING
0620 SBLK1 LDAX D
0621 CPI BLANK
0622 JZ SCHR GOT A BLANK NOW SCAN PAST IT
0623 INX D
0624 DCR C NO MORE THAN TWELVE
0625 JNZ SBLK1
0626 RET GO BACK WITH ZERO FLAG SET
0627 *
0628 *
0629 * SCAN PAST UP TO 10 BLANK POSITIONS LOOKING FOR
0630 * A NON BLANK CHARACTER.
0631 *
0632 SCHR MVI C,10 SCAN TO FIRST NON BLANK CHR WITHIN 10
0633 SCHR1 LDAX D GET NEXT CHARACTER

```

2 OF 3 \*\*\*\*\*

PROGRAM DEVELOPMENT SYSTEM

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 13

```

C24E FE 20      0634      CPI      SPACE
C250 C0         0635      RNZ      .      WE'RE PAST THEM
C251 13         0636      INX      D      NEXT SCAN ADDRESS
C252 0D         0637      DCR      C
C253 C8         0638      RZ      .      COMMAND ERROR
C254 C3 4D C2   0639      JMP      SCHR1  KEEP LOOPING
0640 *
0641 *
0642 *      THIS ROUTINE SCANS OVER CHARACTERS, PAST BLANKS AND
0643 *      CONVERTS THE FOLLOWING VALUE TO HEX.  ERRORS RETURN TO
0644 *      THE ERROR HANDLER.
0645 *
0646 SCONV CALL   SBLK      FIND IF VALUE IS PRESENT
0647      JZ      ERR1      ABORT TO ERROR IF NONE
0648 *
0649 *
0650 *      THIS ROUTINE CONVERTS ASCII DIGITS INTO BINARY FOLLOWING
0651 *      A STANDARD HEX CONVERSION.  THE SCAN STOPS WHEN AN ASCII
0652 *      SPACE IS ENCOUNTERED.  PARAMETER ERRORS REPLACE THE ERROR
0653 *      CHARACTER ON THE SCREEN WITH A QUESTION MARK.
0654 *
0655 SHEX LXI      H,0      CLEAR H & L
0656 SHE1 LDAX    D      GET CHARACTER
0657      CPI      20H      IS IT A SPACE?
0658      RZ      .      IF SO
0659      CPI      '/'      SLASH IS ALSO LEGAL
0660      RZ
0661      CPI      ':'      EVEN THE COLON IS ALLOWED
0662      RZ
0663 *
0664 HCONV DAD      H      MAKE ROOM FOR THE NEW ONE
0665      DAD      H
0666      DAD      H
0667      DAD      H
0668      CALL    HCOV1     DO THE CONVERSION
0669      JNC      ERR1     NOT VALID HEXIDECIMAL VALUE
0670      ADD      L
0671      MOV      L,A      MOVE IT IN
0672      INX      D      BUMP THE POINTER
0673      JMP      SHE1
0674 *
0675 HCOV1 SUI      48      REMOVE ASCII BIAS
0676      CPI      10
0677      RC      .      IF LESS THAN 9
0678      SUI      7      IT'S A LETTER
0679      CPI      10H
0680      RET      .      WITH TEST IN HAND
0681 *
0682 *
0683 *      TERM COMMAND
0684 *
0685 *      THIS ROUTINE GETS CHARACTERS FROM THE SYSTEM KEYBOARD
0686 *      AND OUTPUTS THEM TO THE SERIAL PORT.  IT IS

```

PROGRAM DEVELOPMENT SYSTEM

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 14

```

0687 *      INTENDED TO CONFIGURE THE Sol AS A STANDARD VIDEO
0688 *      TERMINAL.  COMMAND KEYS ARE NOT OUTPUT TO THE OUTPUT
0689 *      PORT BUT ARE INTERPRETED AS DIRECT Sol COMMANDS.
0690 *      THE MODE COMMAND, RECEIVED BY THE KEYBOARD, PUTS THE
0691 *      Sol IN THE COMMAND MODE.
0692 *
0693 *
0694 *
0695 TERM EQU      $      TERM COMMAND VIA PORT 1 ONLY
0696 *
0697 TERM1 CALL    KSTAT   IS THERE ONE WAITING?
0698      JZ      TIN      IF NOT
0699      MOV      B,A      SAVE IT IN B
0700      CPI      MODE    IS IT MODE
0701      JZ      COMN1    YES--RESET AND QUIT TERM
0702      JC      TOUT     NON-CURSOR KEY---SEND TO TERM PORT
0703      CALL    VDMOT    PROCESS IT
0704      JMP      TIN
0705 *
0706 TOUT CALL      SDROT   OUTPUT IT TO THE SERIAL PORT
0707 TIN  CALL      SSTAT   GET INPUT STATUS
0708      JZ      TERM1    LOOP IF NOT
0709      ANI      7FH      NO HIGH BITS FROM HERE
0710      JZ      TERM1    A NULL IS IGNORED
0711      MOV      B,A      IT'S OUTPUT FROM 'B'
0712      CALL    VDMOT    PUT IT ON THE SCREEN
0713      JMP      TERM1    LOOP OVER AND OVER
0714 *
0715 *
0716 *
0717 *      DUMP COMMAND
0718 *
0719 *      THIS ROUTINE DUMPS CHARACTERS FROM MEMORY TO THE
0720 *      CURRENT OUTPUT DEVICE.  ALL VALUES ARE DISPLAYED AS
0721 *      ASCII HEX.
0722 *
0723 *      THE COMMAND FORM IS AS FOLLOWS:
0724 *
0725 *      DUmP addr1 addr2
0726 *
0727 *      THE VALUES FROM ADDR1 TO ADDR2 ARE THEN OUTPUT TO THE
0728 *      OUTPUT DEVICE.  IF ONLY ADDR1 IS SPECIFIED THEN THE
0729 *      VALUE AT THAT ADDRESS IS OUTPUT.
0730 *
0731 DUMP CALL      SCONV   SCAN TO FIRST ADDRESS AND CONVERT IT
0732      PUSH     H      SAVE THE VALUE
0733      CALL    PSCAN   SEE IF SECOND WAS GIVEN
0734      POP      D      GET BACK START
0735      XCHG     .      HL HAS START, DE HAS END
0736 *
0737 DLOOP CALL      CRLP   OUTPUT ADDRESS
0738      CALL    ADOUT   ANOTHER SPACE TO KEEP IT PRETTY
0739      CALL    BOUT

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 15

```

C2C0 0E 10      0740      MVI      C,16  VALUES PER LINE
                  0741 *
C2C2 7E          0742 DLP1  MOV      A,M   GET THE CHR
C2C3 C5          0743      PUSH     B     SAVE VALUE COUNT
C2C4 CD DC C2    0744      CALL     HBOUT  SEND IT OUT WITH A BLANK
C2C7 7D          0745      MOV      A,L   COMPARE DE & HL
C2C8 93          0746      SUB      E
C2C9 7C          0747      MOV      A,H
C2CA 9A          0748      SBB      D
C2CB D2 60 C1    0749      JNC      COMN1  ALL DONE
C2CE C1          0750      POP      B     VALUES PER LINE
C2CF 23          0751      INX      H
C2D0 0D          0752      DCR      C     BUMP THE LINE COUNT
C2D1 C2 C2 C2    0753      JNZ      DLP1  NOT ZERO IF MORE FOR THIS LINE
C2D4 C3 B7 C2    0754      JMP      DLOOP  DO A LPCR BEFORE THE NEXT
                  0755 *
                  0756 *
                  0757 * OUTPUT HL AS HEX 16 BIT VALUE
                  0758 *
C2D7 7C          0759 ADOUT  MOV      A,H   H FIRST
C2D8 CD EF C2    0760      CALL     HEOUT
C2DB 7D          0761      MOV      A,L   THEN "L" FOLLOWED BY A SPACE
                  0762 *
C2DC CD EF C2    0763 HBOUT  CALL     HEOUT
C2DF CD 2E C0    0764      CALL     KSTAT  SEE IF A CHAR WAITING
C2E2 CA EA C2    0765      JZ       BOUT   NO
C2E5 E6 7F      0766      ANI      7FH   SEE IF MODE OR CTL-@
C2E7 CA 60 C1    0767      JZ      COMN1  YES--GET OUT
C2EA 06 20      0768 BOUT   MVI      B,    .
C2EC C3 5B C0    0769      JMP      VDMOT  PUT IT OUT
                  0770 *
C2EF 4F          0771 HEOUT  MOV      C,A   GET THE CHARACTER
C2F0 0F          0772      RRC
C2F1 0F          0773      RRC      MOVE THE HIGH FOUR DOWN
C2F2 0F          0774      RRC
C2F3 0F          0775      RRC
C2F4 CD F8 C2    0776      CALL     HEOU1  PUT THEM OUT
C2F7 79          0777      MOV      A,C   THIS TIME THE LOW FOUR
                  0778 *
C2F8 E6 0F      0779 HEOU1  ANI      OFH   FOUR ON THE FLOOR
C2FA C6 30      0780      ADI      48     WE WORK WITH ASCII HERE
C2FC FE 3A      0781      CPI      58     0-9?
C2FE DA 03 C3    0782      JC      OUTH   YUPI
C301 C6 07      0783      ADI      7     MAKE IT A LETTER
C303 47          0784 OUTH  MOV      B,A   OUTPUT IT FROM REGISTER 'B'
C304 C3 5B C0    0785      JMP      VDMOT
                  0786 *
                  0787 *
                  0788 * ENTER COMMAND
                  0789 *
0790 * THIS ROUTINE GETS VALUES FROM THE KEYBOARD AND ENTERS
0791 * THEM INTO MEMORY. THE INPUT VALUES ARE SCANNED FOLLOWING
0792 * A STANDARD 'GCLIN' INPUT SO ON SCREEN EDITING MAY TAKE

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 16

```

0793 * PLACE PRIOR TO THE LINE TERMINATOR. A BACK SLASH '/'
0794 * ENDS THE ROUTINE AND RETURNS CONTROL TO THE COMMAND MODE.
0795 *
0796 ENTER  CALL     SCONV  SCAN OVER CHARS AND GET ADDRESS
0797      PUSH     H     SAVE ADDRESS
0798 *
0799 ENLOP  CALL     CRLF
0800      MVI      B,    .
0801      CALL     CONT  GET LINE OF INPUT
0802      CALL     CREM  REMOVE THE CURSOR
0803      MVI      C,1   START SCAN
0804      CALL     VDAD2  GET ADDRESS
0805      XCHG      .
0806 *
0807 *
0808 ENLO1  MVI      C,3   NO MORE THAN THREE SPACES BETWEEN VALUES
0809      CALL     SCHR1  SCAN TO NEXT VALUE
0810      JZ       ENLOP  LAST ENTRY FOUND START NEW LINE
0811 *
0812      CPI      '/'    COMMAND TERMINATOR?
0813      JZ      COMN1  IF SO...RETURN TO STANDARD INPUT
0814      CALL     SHEX  CONVERT VALUE
0815      MOV      A,L   GET LOW PART AS CONVERTED
0816      POP      H     GET MEMORY ADDRESS
0817      MOV      M,A   PUT IN THE VALUE
0818      INX      H
0819      PUSH     H     BACK GOES THE ADDRESS
0820      JMP      ENLO1  CONTINUE THE SCAN
0821 *
0822 *
0823 *
0824 * EXECUTE COMMAND
0825 *
0826 * THIS ROUTINE GETS THE FOLLOWING PARAMETER AND DOES A
0827 * PROGRAM JUMP TO THE LOCATION GIVEN BY IT. IF PROPER
0828 * STACK OPERATIONS ARE USED WITHIN THE EXTERNAL PROGRAM
0829 * IT CAN DO A STANDARD 'RETURN' TO THE SOLOS COMMAND MODE.
0830 * THE STARTING ADDRESS OF SOLOS IS PASSED TO THE PROGRAM
0831 * IN REGISTER PAIR HL SO IT CAN ADJUST INTERNAL PARAMETERS
0832 * FOR SOLOS OPERATION.
0833 *
0834 *
0835 EXEC  CALL     SCONV  SCAN PAST BLANKS AND GET PARAMETER
0836 EXEC1  PUSH     H     PUT GO ADDRESS ON STACK
0837      LXI      H,START  TELL THE PROGRAM WHERE WE CAME FROM
0838      RET      .     AND DISPATCH TO IT
0839 *
0840 *
0841 *
0842 * SOLOS ERROR HANDLER
0843 *
0844 ERR1  XCHG      .
0845 ERR2  MVI      M,'?'  PUT QUESTION MARK ON SCREEN

```



PROGRAM DEVELOPMENT SYSTEM

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 17

```

C33F C3 60 C1      0846      JMP      COMN1 AND RETURN TO COMMAND MODE
                   0847 *
                   0848 *
                   0849 *
                   0850 * --
                   9999      COPY      CONSOL3/1
                   0851 *
                   0852 *
                   0853 * THIS ROUTINE PROCESSES THE "GET" COMMAND
                   0854 *
C342               0855 TLOAD EQU $      PREP TO GET THE VERY NEXT FILE
C342 3A 0D C8      0856 LDA TSPD GET SPEED FM SYSTEM RAM AREA
C345 F6 C0         0857 ORI TAPE1-TAPE2 TURN BOTH MACHINES ON
                   0858 * DROP THRU TO READ IN THIS FILE
                   0859 *
                   0860 * TAPE READ ROUTINE
                   0861 *
                   0862 * ON ENTRY: A HAS UNIT AND SPEED
                   0863 *
                   0864 * ON EXIT: IF CARRY WAS SET---BELL TO SCREEN
                   0865 * OTHERWISE--NORMAL--BACK TO COMN1
                   0866 *
                   0867 *
C347               0868 RTAPE EQU $      READ VERY NEXT TAPE FILE
C347 06 03         0869 MVI B,3 SHORT DELAY
C349 CD EE C3      0870 CALL TON
C34C DB FB         0871 IN TDATA CLEAR THE UART FLAGS
                   0872 *
C34E               0873 PTAP1 EQU $      HERE TO RD THE HDR
C34E CD 8E C3      0874 CALL RHEAD GO READ HEADER
C351 DA 78 C3      0875 JC TERR IF AN ERROR OR ESC WAS RECEIVED
C354 C2 4E C3      0876 JNZ PTAP1 IF VALID HEADER NOT FOUND
                   0877 *
                   0878 * FOUND A VALID HEADER NOW JUST READ THE FILE IN
                   0879 *
C357 2A 23 C8      0880 LHLD BLOCK GET BLOCK SIZE
C35A EB           0881 XCHG . ...TO DE
C35B 2A 25 C8      0882 LHLD LOADR GET TAPE LOAD ADDRESS
                   0883 *
                   0884 *
                   0885 * THIS ROUTINE READS "DE" BYTES FROM THE TAPE
                   0886 * TO ADDRESS HL. THE BYTES MUST BE FROM ONE
                   0887 * CONTIGUOUS PHYSICAL BLOCK ON THE TAPE.
                   0888 *
                   0889 * HL HAS "PUT" ADDRESS
                   0890 * DE HAS SIZE OF TAPE BLOCK
                   0891 *
C35E               0892 RTAP EQU $      ACTUALLY RD IT NOW
C35E           0893 *
C35E               0894 RTAP2 EQU $      HERE TO LOOP RDING BLKS
C35E CD 80 C3      0895 CALL DCRCT DROP COUNT, B=LEN THIS BLK
C361 CA 75 C3      0896 JZ RTOFF ZERO=ALL DONE
                   0897 *

```

3 OF 3 \*\*\*\*\*

PROGRAM DEVELOPMENT SYSTEM

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 18

```

C364 CD AF C3      0898      CALL RHED1 READ THAT MANY BYTES
C367 DA 78 C3      0899 JC TERR IF ERROR OR MODE STOPPED US
C36A CA 5E C3      0900 JZ RTAP2 RD OK--READ SOME MORE
C36D C3 78 C3      0901 JMP TERR CRC ERROR--TAKE ERROR EXIT
                   0902 *
                   0903 *
                   0904 *
C370 06 01         0905 TOFF MVI B,1
C372 CD F0 C3      0906 CALL DELAY
C375             0907 RTOFF EQU $      TURN OFF TAPES AND EXIT
C375 D2 60 C1      0908 JNC COMN1 WENT AOK, WE'RE DONE
C378             0909 TERR EQU $      HERE WHEN AN ERROR IS TO STOP US
C378 06 07         0910 MVI B,'G'-40H WE HAD AN ERROR--PUT BELL ON SCREEN
C37A CD 5B C0      0911 CALL VDMOT
C37D C3 60 C1      0912 JMP COMN1 DONE AND TURN TAPES OFF
                   0913 *
                   0914 *
C380             0915 DCRCT EQU $      COMMON RTN TO COUNT DOWN BLK LENGTHS
C380 AF           0916 XRA A CLR FOR LATER TESTS
C381 47           0917 MOV B,A SET THIS BLK LEN=256
C382 B2           0918 ORA D IS AMNT LEFT < 256
C383 C2 8B C3      0919 JNZ DCRCT2 NO--REDUCE AMNT BY 256
C386 B3           0920 ORA E IS ENTIRE COUNT ZERO
C387 C8           0921 RZ ALL DONE--ZERO-THIS CONDITION
C388 43           0922 MOV B,E SET THIS BLK LEN TO AMNT REMAINING
C389 5A           0923 MOV E,D MAKE ENTIRE COUNT ZERO NOW
C38A C9           0924 RET ALL DONE (NON-ZERO FLAG)
C38B             0925 DCRCT2 EQU $      REDUCE COUNT BY 256
C38B 15           0926 DCR D DROP BY 256
C38C B7           0927 ORA A FORCE NON-ZERO FLAG
C38D C9           0928 RET NON-ZERO=NOT DONE YET (BLK LEN=256)
                   0929 *
                   0930 *
                   0931 * READ THE HEADER
                   0932 *
C38E 06 0A         0933 RHEAD MVI B,10 FIND 10 NULLS
C390 CD C8 C3      0934 RHEAD1 CALL STAT
C393 D8           0935 RC . IF ESCAPE
C394 DB FB         0936 IN TDATA IGNORE ERROR CONDITIONS
C396 B7           0937 ORA A ZERO?
C397 C2 8E C3      0938 JNZ RHEAD
C39A 05           0939 DCR B
C39B C2 90 C3      0940 JNZ RHEAD1 LOOP UNTIL 10 IN A ROW
                   0941 *
                   0942 * WAIT FOR THE START CHARACTER
                   0943 *
C39E CD DA C3      0944 SOHL CALL TAPIN
C3A1 D8           0945 RC . ERROR OR ESCAPE
C3A2 FE 01         0946 CPI 1 LOOK FOR SOH (HEX 01)
C3A4 DA 9E C3      0947 JC SOHL ZERO--STILL WAITING FOR A ONE
C3A7 C2 8E C3      0948 JNZ RHEAD NOT ZERO OR ONE--LK FOR ANOTHER 10 NULLS
                   0949 *
                   0950 * NOW GET THE HEADER

```

PROGRAM DEVELOPMENT SYSTEM

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 19

```

0951 *
C3AA 21 1C C8 0952 LXI H,THEAD POINT TO BUFFER
C3AD 06 10 0953 MVI B,HLEN LENGTH TO READ
0954 *
C3AF 0E 00 0955 RHED1 EQU $ RD A BLOCK INTO HL FOR B BYTES
C3B1 0E 00 0956 MVI C,0 INIT THE CRC
C3B1 CD DA C3 0957 RHED2 EQU $ LOOP HERE
C3B4 D8 0958 CALL TAPIN GET A BYTE
C3B5 77 0959 RC
C3B6 23 0960 MOV M,A STORE IT
C3B7 CD E7 C3 0961 INX H INCREMENT ADDRESS
C3BA 05 0962 CALL DOCR C GO COMPUTE THE CRC
C3BB C2 B1 C3 0963 DCR B WHOLE HEADER YET?
0964 JNZ RHED2 DO ALL THE BYTES
0965 *
0966 * THIS ROUTINE GETS THE NEXT BYTE AND COMPARES IT
0967 * TO THE VALUE IN REGISTER C. THE FLAGS ARE SET ON
0968 * RETURN.
0969 *
C3BE CD DA C3 0970 CALL TAPIN GET CRC BYTE
C3C1 A9 0971 XRA C CLR CARRY AND SET ZERO IF MATCH, ELSE NON-ZERO
C3C2 C8 0972 RZ CRC OK
C3C3 3A 11 C8 0973 LDA IGNCR GET CRC IGNORE FLAG
C3C6 3C 0974 INR A SET FLAGS SO THAT CRC ERRORS CAN BE IGNORED IF FF
C3C7 C9 0975 RET
0976 *
0977 * THIS ROUTINE GETS THE NEXT AVAILABLE BYTE FROM THE
0978 * TAPE. WHILE WAITING FOR THE BYTE THE KEYBOARD IS TESTED
0979 * FOR AN ESC COMMAND. IF RECEIVED THE TAPE LOAD IS
0980 * TERMINATED AND A RETURN TO THE COMMAND MODE IS MADE.
0981 *
C3C8 DB FA 0982 STAT IN TAPPT TAPE STATUS PORT
C3CA E6 40 0983 ANI TDR
C3CC C0 0984 RNZ
C3CD CD 2E C0 0985 CALL KSTAT CHECK SOL KBD
C3D0 CA C8 C3 0986 JZ STAT NOTHING THERE YET
C3D3 E6 7F 0987 ANI 7FH CLR PARITY 1ST
C3D5 C2 C8 C3 0988 JNZ STAT NOT A MODE (OR EVEN CTL-0)
C3D8 37 0989 STC SET ERROR FLAG
C3D9 C9 0990 RET AND RETURN
0991 *
0992 *
0993 *
C3DA CD C8 C3 0994 TAPIN CALL STAT WAIT UNTIL A CHARACTER IS AVAILABLE
C3DD D8 0995 RC
0996 *
C3DE DB FA 0997 TREDY IN TAPPT TAPE STATUS
C3E0 E6 18 0998 ANI TFE+TOE DATA ERROR?
C3E2 DB FB 0999 IN TDATA GET THE DATA
C3E4 C8 1000 RZ IF NO ERRORS
C3E5 37 1001 STC SET ERROR FLAG
C3E6 C9 1002 RET
1003 *

```

PROGRAM DEVELOPMENT SYSTEM

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 20

```

1004 *
C3E7 0955 DOCR EQU $ A COMMON CRC COMPUTATION ROUTINE
C3E7 91 1006 SUB C
C3E8 4F 1007 MOV C,A
C3E9 A9 1008 XRA C
C3EA 2F 1009 CMA
C3EB 91 1010 SUB C
C3EC 4F 1011 MOV C,A
C3ED C9 1012 RET ONE BYTE NOW WRITTEN
1013 *
1014 *
C3EE D3 FA 1015 TON EQU $ HERE TO TURN A TAPE ON THEN DELAY
1016 OUT TAPPT GET TAPE MOVING, THEN DELAY
1017 *
C3F0 11 00 00 1018 DELAY LXI D,0
C3F3 1B 1019 DLOP1 DCX D
C3F4 7A 1020 MOV A,D
C3F5 B3 1021 ORA E
C3F6 C2 F3 C3 1022 JNZ DLOP1
C3F9 05 1023 DCR B
C3FA C2 F0 C3 1024 JNZ DELAY
C3FD C9 1025 RET
1026 *
1027 *
1028 * *****END OF PROGRAM*****
1029 *
1030 *
1031 *
1032 *
1033 * SYSTEM EQUATES
1034 *
1035 *
1036 * VDM PARAMETERS
1037 *
CC00 1038 VDMEM EQU 0CC00H VDM SCREEN MEMORY
1039 *
1040 *
1041 *
1042 * KEYBOARD SPECIAL KEY ASSIGNMENTS
1043 *
1044 * THESE DEFINITIONS ARE DESIGNED TO ALLOW
1045 * COMPATABILITY WITH CUTER(TM). THESE ARE THE
1046 * SAME KEYS AS BELOW, EXCEPT THAT THE X'80' BIT
1047 * (BIT 7) IS STRIPPED OFF.
1048 *
1048 DOWN EQU 9AH (CTL-Z)
1049 UP EQU 97H (CTL-W)
1050 LEFT EQU 81H (CTL-A)
1051 RIGHT EQU 93H (CTL-S)
1052 CLEAR EQU 8BH (CTL-K)
1053 HOME EQU 8EH (CTL-N)
1054 MODE EQU 80H (CTL-0)
1055 BACKS EQU 5FH BACKSPACE
1056 LF EQU 10

```

PROGRAM DEVELOPMENT SYSTEM

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 21

```

000D 1057 CR EQU 13
0020 1058 BLANK EQU
0020 1059 SPACE EQU BLANK
0018 1060 CX EQU 'X'-40H
001B 1061 ESC EQU 1BH
1062 *
1063 *
1064 *
1065 STAPT EQU OFAH STATUS PORT GENERAL
1066 SERST EQU OF8H SERIAL STATUS PORT
1067 SDATA EQU OF9H SERIAL DATA
1068 PDATA EQU OFDH PARALLEL DATA
1069 KDATA EQU OFCH KEYBOARD DATA
1070 DSTAT EQU OFEH VDM CONTROL PORT
1071 TAPPT EQU OFAH TAPE STATUS PORT
1072 TDATA EQU OFBH TAPE DATA PORT
1073 SENSE EQU OFFH SENSE SWITCHES
1074 *
1075 *
1076 *
1077 *
1078 *
1079 SCD EQU 1 SERIAL CARRIER DETECT
1080 SDSR EQU 2 SERIAL DATA SET READY
1081 SPE EQU 4 SERIAL PARITY ERROR
1082 SFE EQU 8 SERIAL FRAMING ERROR
1083 SOE EQU 16 SERIAL OVERRUN ERROR
1084 SCTS EQU 32 SERIAL CLEAR TO SEND
1085 SDR EQU 64 SERIAL DATA READY
1086 STBE EQU 128 SERIAL TRANSMITTER BUFFER EMPTY
1087 *
1088 KDR EQU 1 KEYBOARD DATA READY
1089 PDR EQU 2 PARALLEL DATA READY
1090 PXDR EQU 4 PARALLEL DEVICE READY
1091 TFE EQU 8 TAPE FRAMING ERROR
1092 TOE EQU 16 TAPE OVERFLOW ERROR
1093 TDR EQU 64 TAPE DATA READY
1094 TTBE EQU 128 TAPE TRANSMITTER BUFFER EMPTY
1095 *
1096 SOK EQU 1 SCROLL OK FLAG
1097 *
1098 TAPE1 EQU 80H 1=TURN TAPE ONE ON
1099 TAPE2 EQU 40H 1=TURN TAPE TWO ON
1100 *
1101 *
1102 *
1103 *
1104 *
1105 *
1106 *
1107 *
1108 SYSRAM EQU $ START OF SYSTEM RAM
1109 SYSTP EQU SYSRAM+3FFH STACK WORKS FM TOP DOWN

```

PROGRAM DEVELOPMENT SYSTEM

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 22

```

1110 *
1111 *
1112 *
1113 *
1114 *
1115 *
1116 *
1117 *
1118 *
1119 UIPRT DS 2 USER DEFINED INPUT RTN IF NON ZERO
1120 UOPRT DS 2 USER DEFINED OUTPUT RTN IF NON ZERO
1121 DPLTS DS 2 DEFAULT PSUEDO I/O PORTS (ALWAYS ZERO IN SOLOS)
1122 IPORT DS 1 CRMT INPUT PSUEDO PORT
1123 OPORT DS 1 CRMT OUTPUT PSUEDO PORT
1124 NCHAR DS 1 CURRENT CHARACTER POSITION
1125 LINE DS 1 CURRENT LINE POSITION
1126 BOT DS 1 BEGINNING OF TEXT DISPLACEMENT
1127 SPEED DS 1 SPEED CONTROL BYTE
1128 ESCFL DS 1 FOR COMPATABILITY W/ SOLOS/CUTER
1129 TSPD DS 1 CURRENT TAPE SPEED---00=FAST, 20H=SLOW
1130 INPTR DS 2 FOR COMPATABILITY W/ CUTER
1131 NUCNT DS 1 FOR COMPATABILITY W/ SOLOS/CUTER
1132 IGNCR DS 1 FF=IGNORE CRC ERRORS, ELSE=NORMAL
1133 *
1134 DS 10 ROOM FOR FUTURE EXPANSION
1135 *
1136 *
1137 *
1138 *
1139 *
1140 THEAD DS 5 NAME
1141 DS 1 THIS BYTE MUST BE ZERO
1142 HTYPE DS 1 TYPE
1143 BLOCK DS 2 BLOCK SIZE
1144 LOADR DS 2 LOAD ADDRESS
1145 XEQAD DS 2 AUTO EXECUTE ADDRESS
1146 HSPR DS 3 SPARES
1147 *
1148 HLEN EQU $-THEAD LENGTH OF HEADER
1149 BLKOF EQU BLOCK-THEAD OFFSET TO BLOCK SIZE
1150 DHEAD DS HLEN A DUMMY HDR FOR COMPARES WHILE RD'ING
1151 *
1152 *
1153 CUTAB DS 6*4 FOR COMPATABILITY W/ SOLOS/CUTER
1154 *
1155 *
1156 FNUMF DS 1 FOR CURRENT FILE OPERATIONS
1157 FCBAS DS 7 1ST FILE CONTROL BLOCK
1158 FCBA2 DS 7 2ND FILE CONTROL BLOCK
1159 FBUFF1 DS 2*256 SYSTEM FILE BUFFER BASE
1160 DS 81 THIS IS AN AREA USED BY CUTER
1161 USARE EQU $ START OF USER AREA
1162 *

```

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 23

1163 \* REMEMBER THAT THE STACK WORKS ITS WAY DOWN FROM  
1164 \* THE END OF THIS 1K RAM AREA.  
1165 \*  
1166 \*  
1167 \* --

ADOUT	C2D7	AINP	C022	AOUT	C01C	BACKS	005F
BLANK	0020	BLKOF	0007	BLOCK	C823	BOT	C80A
BOUT	C2EA	CHAR	C08A	CHPCK	C05E	CLEAR	008B
CLERA	C154	CLIN1	C0F0	CLINE	C0EA	COMN1	C160
COMND	C169	COMTA	C1DE	CONT	C193	COPRC	C199
CR	000D	CREM	C12C	CRLF	C22B	CUR	C0C7
CURSC	C0C5	CUTAB	C83C	CX	0018	DCRC2	C38B
DCRCT	C380	DELAY	C3F0	DFLTS	C804	DHEAD	C82C
DISP1	C1BC	DISPT	C1B8	DLOOP	C2B7	DLOP1	C3F3
DLP1	C2C2	DOCRC	C3E7	DOWN	009A	DSTAT	00FE
DUMP	C2AE	ENLO1	C31C	ENLOP	C30R	ENTER	C307
ERAS1	C0D1	ERAS3	C0E4	ERR1	C33C	ERR2	C33D
ERRCT	C160	ESC	001B	ESCFL	C80C	EXEC	C334
EXEC1	C337	FBUF1	C863	FCBA2	C85C	FCBAS	C855
EXLOS	C00A	FDCOM	C1C5	FNUMF	C854	FOPEN	C007
GCLIN	C178	GOBAC	C06B	GOBAS	C1BF	GORK	C072
HBOU1	C2DC	HCONV	C26A	HCOV1	C27A	HEOU1	C2F8
HEOUT	C2EF	HLEN	0010	HOME	008E	HSPR	C829
HTYPE	C822	IGNCR	C811	INIT	C001	INPTR	C80E
IOPRC	C026	IPOINT	C806	ITAB	C21B	KDATA	00FC
KDR	0001	KSTAT	C02E	LEFT	0081	LF	000A
LINE	C809	LOADR	C825	MODE	0080	NCHAR	C808
NCOM	C1D7	NEXT	C076	NUCNT	C810	OCHAR	C08E
OK	C0B7	OPOINT	C807	OTAB	C213	OUTH	C303
OUTPR	C03B	PBACK	C134	PCR	C13D	PCUR	C105
PDATA	00FD	PDOWN	C0C1	PDR	0002	PERSE	C0CB
PHOME	C0DB	PLEFT	C101	PLF	C143	PRIT	C10B
PROMP	C223	PSCAN	C235	PTAP1	C34E	PUP	C0FA
PXDR	0004	RDBLK	C013	RDBYT	C00D	RETRN	C004
RHEA1	C390	RHEAD	C38E	RHED1	C3AF	RHED2	C3B1
RIGHT	0093	RTAP	C35E	RTAP2	C35E	RTAPE	C347
RTOFF	C375	SBLK	C23D	SBLK1	C23F	SCD	0001
SCHR	C24B	SCHR1	C24D	SCONV	C257	SCROL	C0A2
SCTS	0020	SDATA	00F9	SDR	0040	SDROT	C04A
SDSR	0002	SENSE	00FF	SERST	00F8	SFE	0008
SHE1	C260	SHEX	C25D	SINP	C01F	SOE	0010
SOHL	C39E	SOK	0001	SOUT	C019	SPACE	0020
SPE	0004	SPEED	C80B	SROL	C0A6	SSTAT	C042
STAPT	00FA	START	C000	STAT	C3C8	STBE	0080
STRTA	C14F	SYSRA	C800	SYSTP	CBFF	TAPE1	0080
TAPE2	0040	TAFIN	C3DA	TAPPT	00FA	TBL	C1F7
TDATA	00FB	TDR	0040	TERM	C284	TERM1	C284
TERR	C378	TFE	0008	THEAD	C81C	TIN	C29C
TLOAD	C342	TOE	0010	TOFF	C370	TON	C3EE
TOUT	C299	TREDY	C3DE	TSPD	C80D	TSRCH	C078
TTBE	0080	UIPRT	C800	UOPRT	C802	UP	0097
USARE	CAB4	VDAD	C119	VDAD2	C116	VDADD	C112

\*\* PROGRAM DEVELOPMENT SYSTEM \*\*

CONSOL (TM) 77-04-23  
COPYRIGHT (C) 1976, 1977

SOFTWARE TECHNOLOGY CORP.  
P.O. BOX 5260  
SAN MATEO, CA 94402

PAGE 24

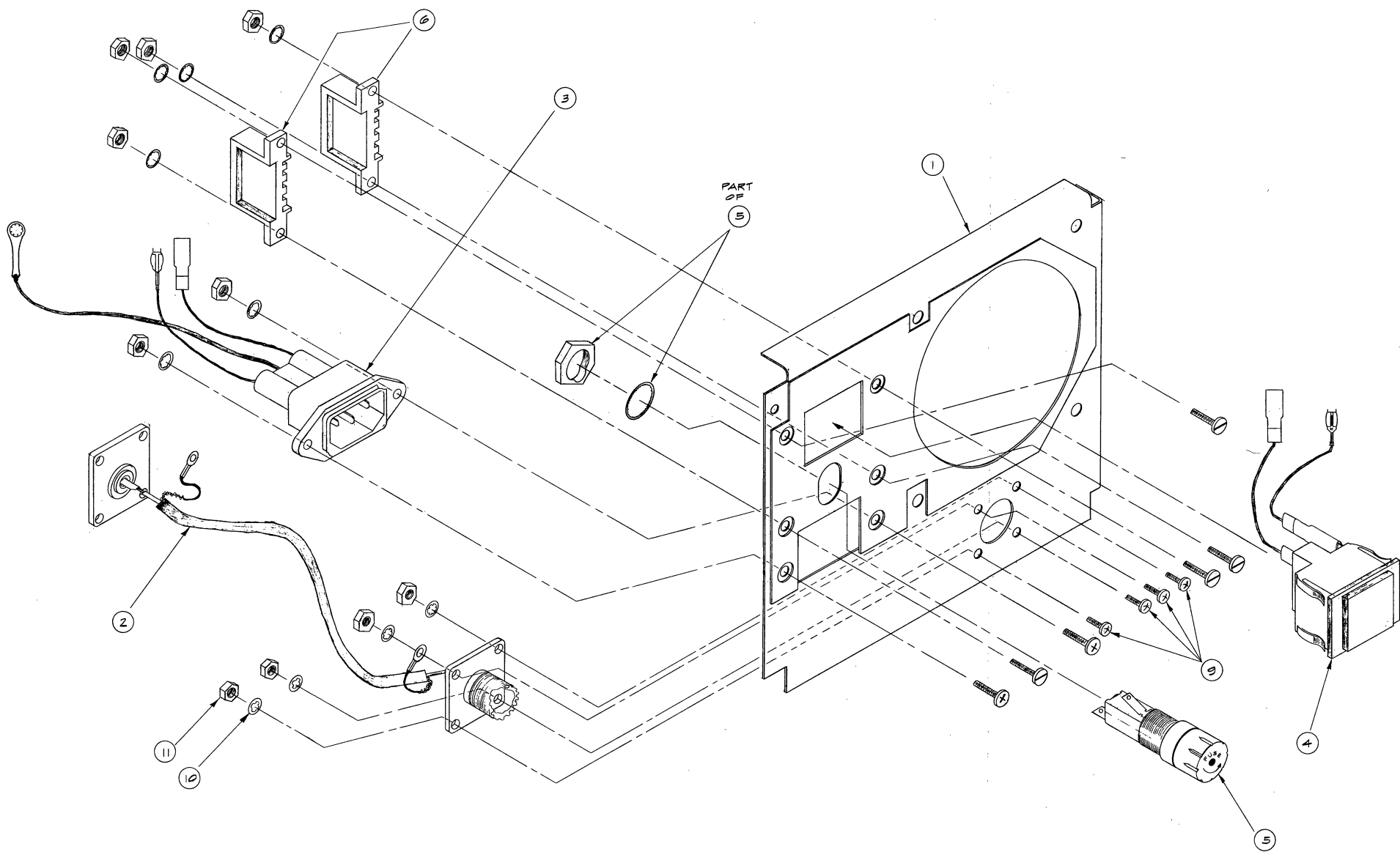
VDMEM CC00 VDMOT C05B VDMOX C054 WRBLK C016  
WRBYT C010 XEQAD C827



X DRAWINGS

- X-1 Assembly, Fan Closure Plate, Drawing #105014
- X-2 Assembly, Sol-10 & 20 Power Supply, Drawing #105001
- X-3 Sol-PC Assembly Drawing
- X-4 Personality Module Assembly Drawing, PM5204
- X-5 Personality Module Assembly Drawing, PM6834
- X-6 Personality Module Assembly Drawing, PM2708
- X-7 Sol Keyboard Assembly Drawing
- X-8 Side Assembly, Left-hand, Drawing #101007
- X-9 Side Assembly, Right-hand, Drawing #101008
- X-10 Assembly, Sol, Drawing #101000 (Sheet 1)
- X-11 Assembly, Sol, Drawing #101000 (Sheet 2)
- X-12 Sol-REG Schematic Drawing
- X-13 Sol-10 Schematic Drawing
- X-14 Sol-20 Schematic Drawing
- X-15 Sol CPU and Bus Schematic, Drawing 1
- X-16 Sol Memory and Decoder Schematic, Drawing 2
- X-17 Sol Input/Output Schematic, Drawing 3
- X-18 Sol Display Control Schematic, Drawing 4
- X-19 Sol Audio Tape I/O Schematic, Drawing 5
- X-20 Personality Module (PM5204) Schematic
- X-21 Personality Module (PM6834) Schematic
- X-22 Personality Module (PM2708) Schematic
- X-23 Sol Keyboard Schematic
- X-24 Sol-PC Block Diagram
- X-25 Sol-Keybaord Block Diagram
- X-26 Sol-Keybaord Photo

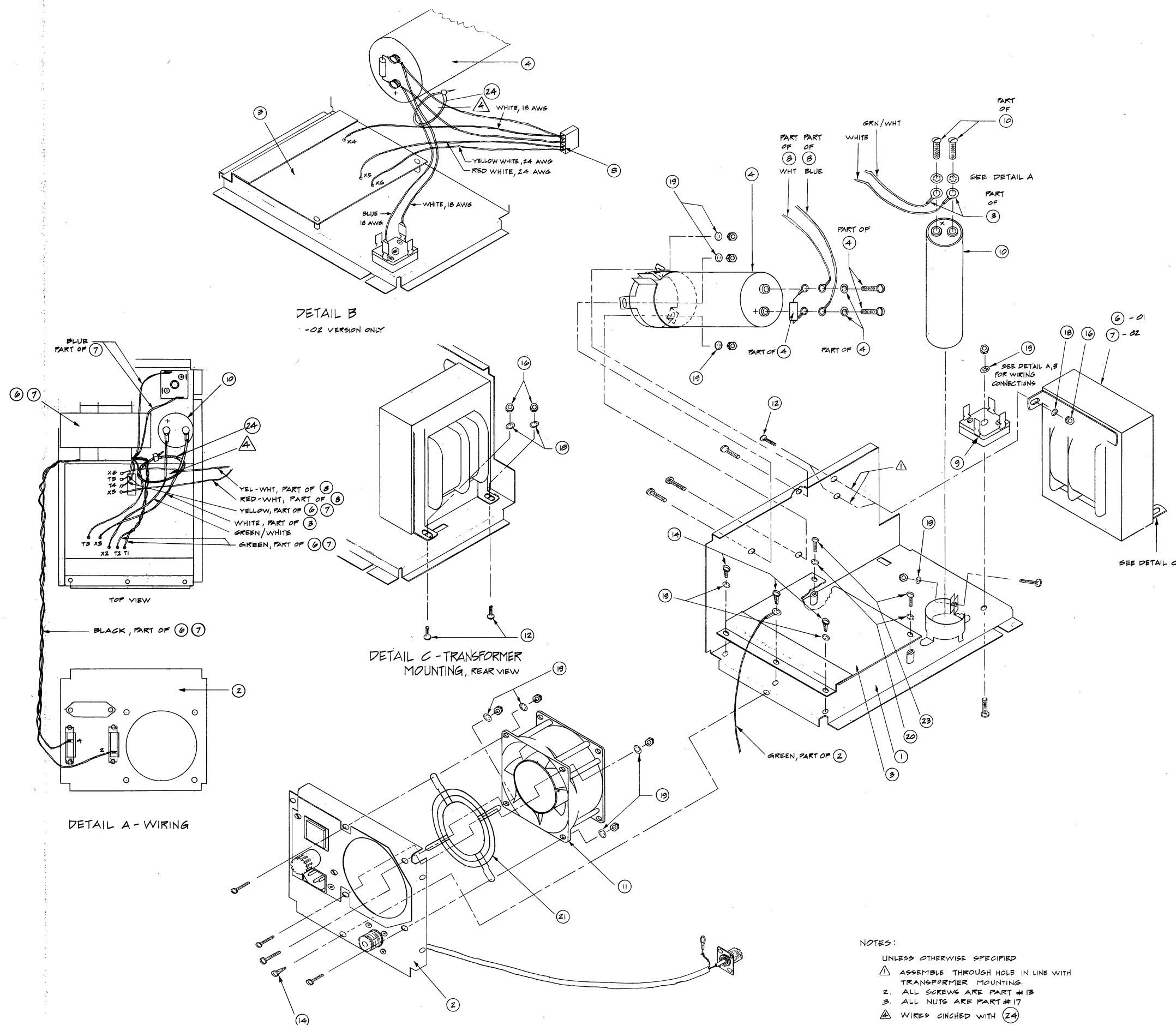
- NOTES:
- UNLESS OTHERWISE SPECIFIED:
- 1. ALL SCREWS PART NO. ⑦
  - 2. ALL LOCKWASHERS PART NO. ⑫
  - 3. ALL NUTS PART NO. ⑧



12	LOCKWASHER, INT. TOOTH # 6	6
11	NUT, HEX, 4-40	4
10	LOCKWASHER, INT. TOOTH # 4	3
9	SCREW, MACHINE, 4-40 x 3/16 PAN HEAD	4
8	NUT, HEX, 6-32	6
7	SCREW, MACHINE, 6-32 x 1/2 PAN HEAD	6
6	COMMONING BLOCKS	2
5	FUSE POST, W/ NUT & LOCKWASHER	1
4	105022 ASSEMBLY, AC SWITCH	1
3	105018 ASSEMBLY, AC CONNECTOR	1
2	105016 ASSEMBLY, VIDEO CABLE	1
1	105031 SHEET METAL, FAN CLOSURE	1
REF. DES.	PART OR DWG. NO.	PART DESCRIPTION
		-01

ASSEMBLY, FAN CLOSURE		
SCALE: —	APPROVED BY: R. MARSH	DRAWN BY LITO
DATE: 01-3-77	REVISOR	REVISED
PROCESSOR TECHNOLOGY		
DRAWING NUMBER		105014





24		TIE, CABLE, PLASTIC	1	2
23		SCREW, 4-40 x 7/16 PAN HEAD	2	2
22	105002	SCHEMATIC, 501-PWR, 501-20	1	REF
21	105003	SCHEMATIC, 501-PWR, 501-10	REF	2
20		LOCKWASHER, #4 INT. TOOTH	2	2
19		LOCKWASHER, #6 INT. TOOTH		
18		LOCKWASHER, #8 INT. TOOTH	3	3
17		NUT, 6-32 HEX		
16		NUT, 8-32 HEX	3	3
15		SCREW, 4-40 x 5/16 PAN HEAD	2	2
14		SCREW, #6 x 5/16 SELF TAPPING PAN HEAD	4	4
13		SCREW, 6-32 x 1/2 PAN HEAD		
12		SCREW, 8-32 x 1/2 PAN HEAD	3	3
11		FAN	0	1
10		CAPACITOR, ALUM. 10,000 MF, 12V	1	1
9		RECTIFIER BRIDGE	0	1
8	105007	CABLE ASSEMBLY, BFD DC PWR	0	1
7	105028	TRANSFORMER, 501-20	0	1
6	105026	TRANSFORMER, 501-10	1	0
5		FINGER GUARD, PAN	0	1
4	105005	CAPACITOR ASSEMBLY, 50,000 MF	0	1
3	105008	REGULATOR ASSEMBLY	1	1
2	105014	PAN CLOSURE ASSEMBLY	1	1
1	103004	CHASSIS ASSEMBLY	1	1
PART OR DWG. NO.			-01	-02
PART DESCRIPTION				

NOTES:

UNLESS OTHERWISE SPECIFIED

- 1. ASSEMBLE THROUGH HOLE IN LINE WITH TRANSFORMER MOUNTING.
- 2. ALL SCREWS ARE PART # 13
- 3. ALL NUTS ARE PART # 17
- 4. WIRES CINCED WITH (24)

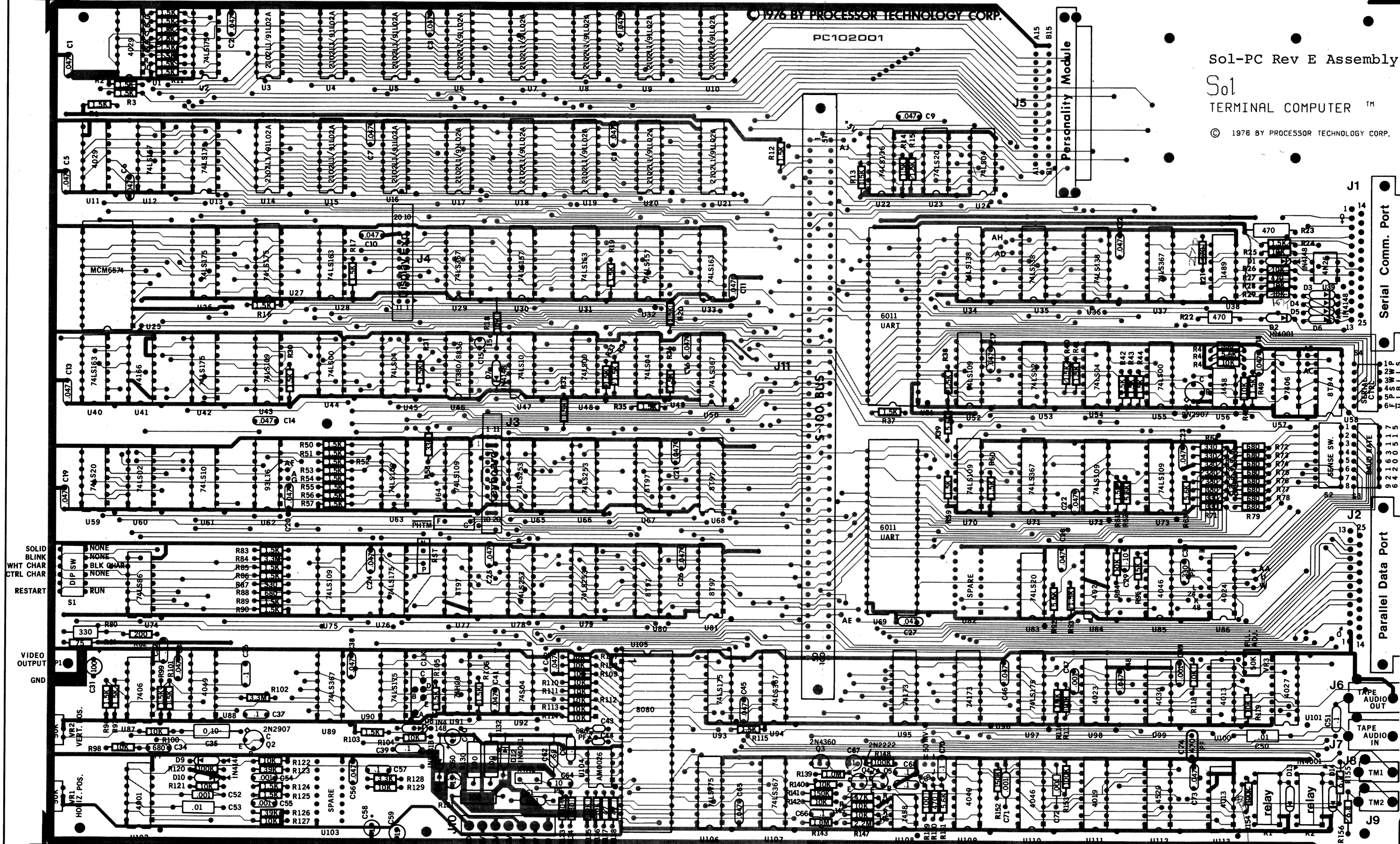
ASSEMBLY, POWER SUPPLY, SOL-PWR		
SCALE: _____	APPROVED BY: R. MARSH	DRAWN BY LITO
DATE: 12-29-76		REVISED
PROCESSOR TECHNOLOGY CORP.		
(501 10 2 501 20) 105001-01 & 105001-02		DRAWING NUMBER 105001



Sol-PC Rev E Assembly

Sol  
TERMINAL COMPUTER™

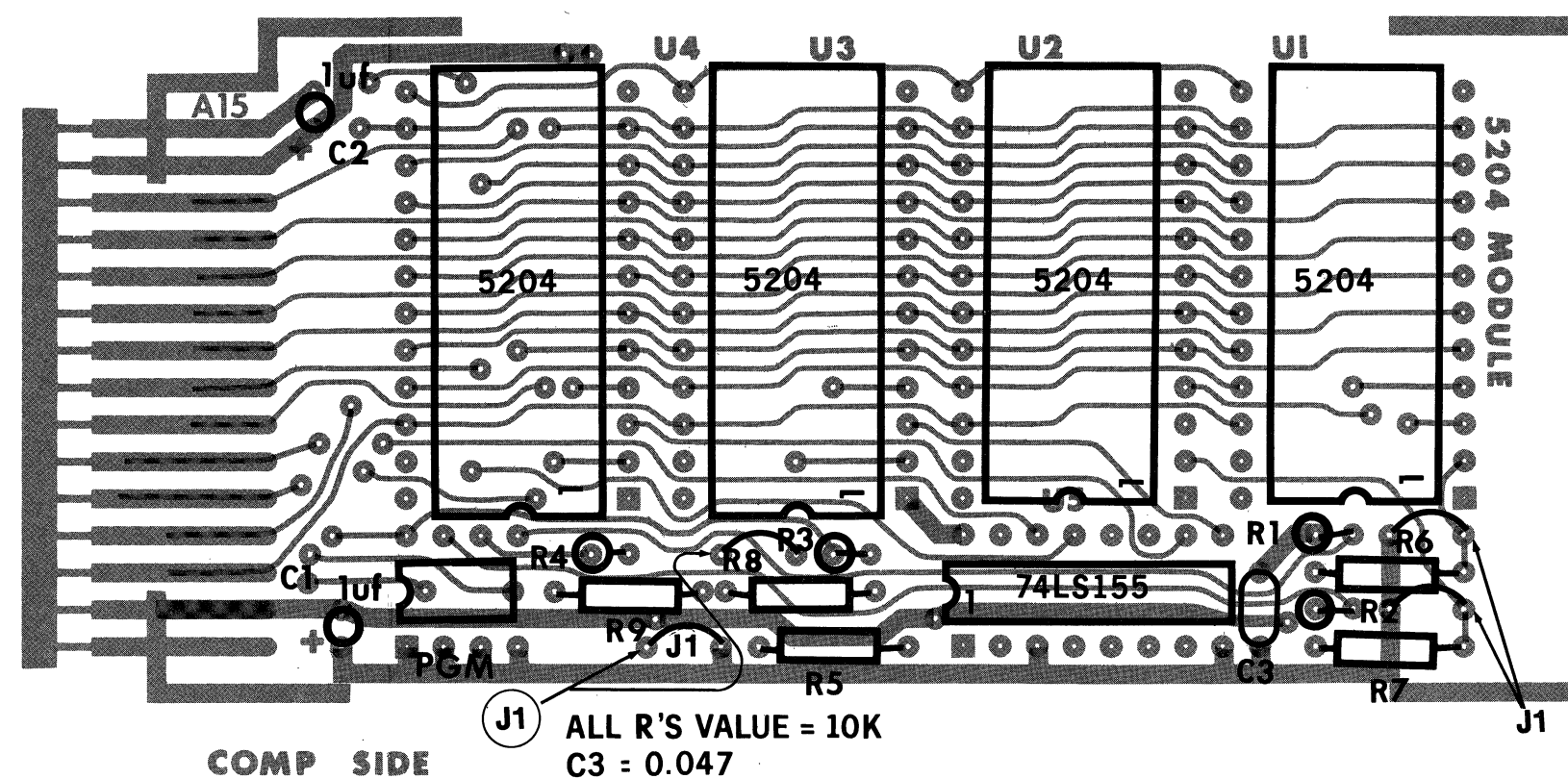
© 1976 BY PROCESSOR TECHNOLOGY CORP.

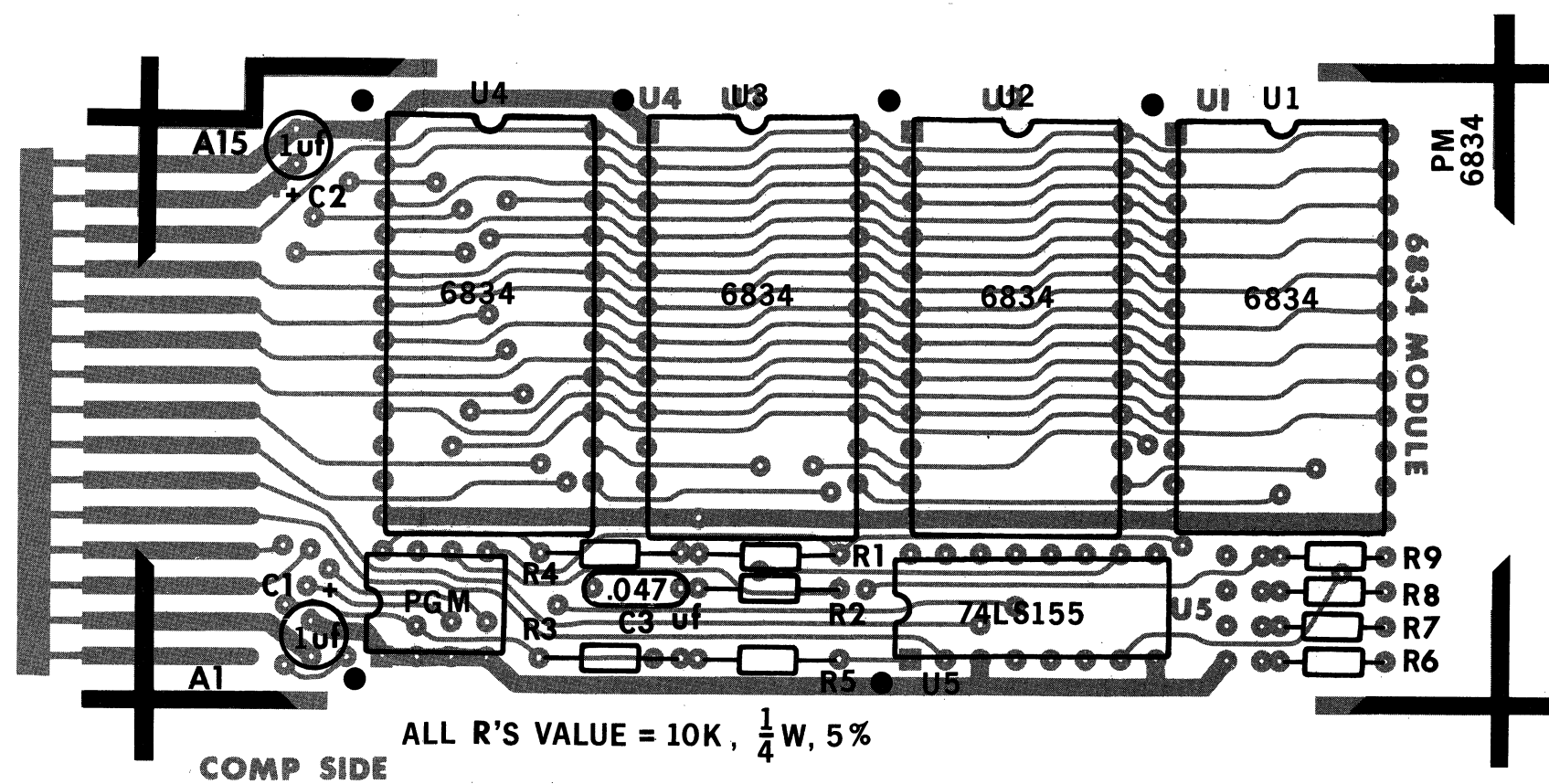


NOTE:  
R91 is not used

VR-3  
100K

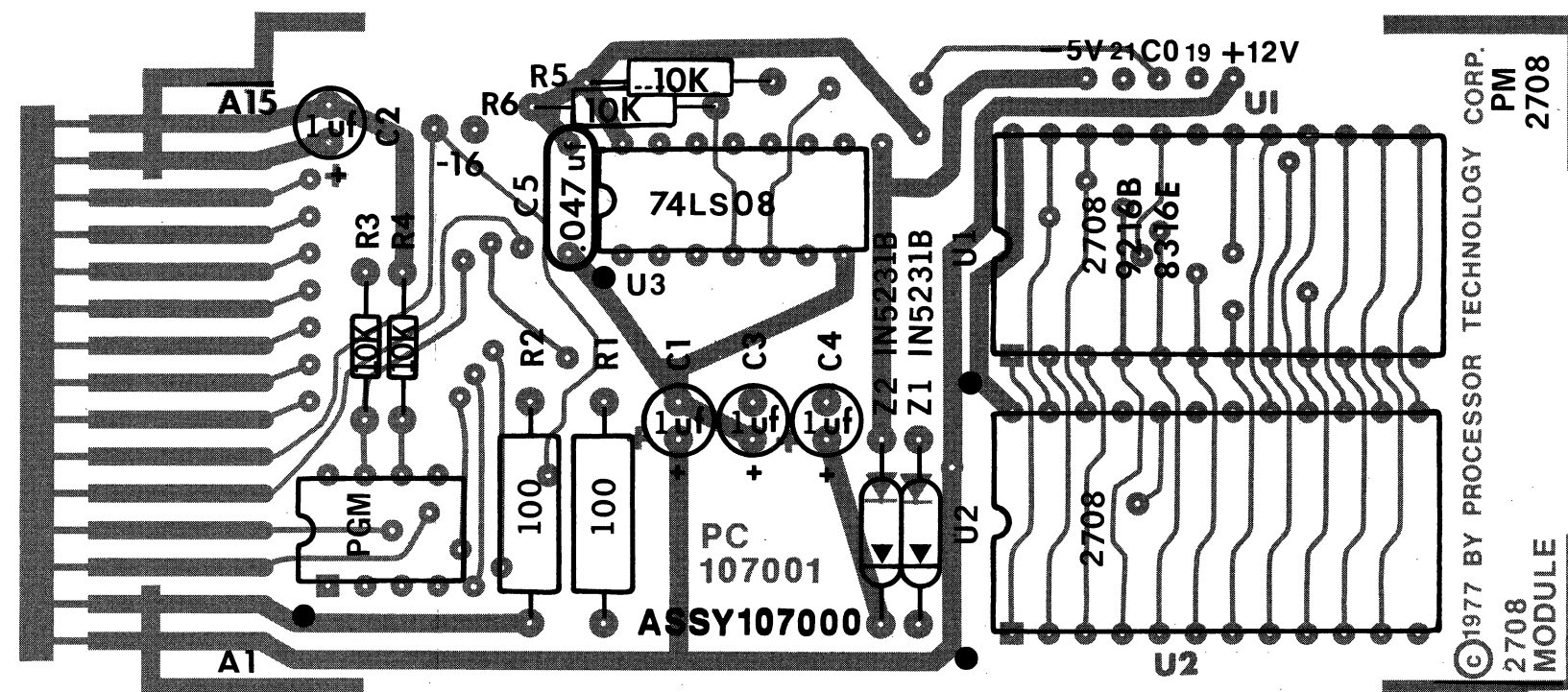
Note: jumpers labelled "J1" must be installed on PM5204 modules using National 5204Q PROMS.





PM 6834

ASSEMBLY



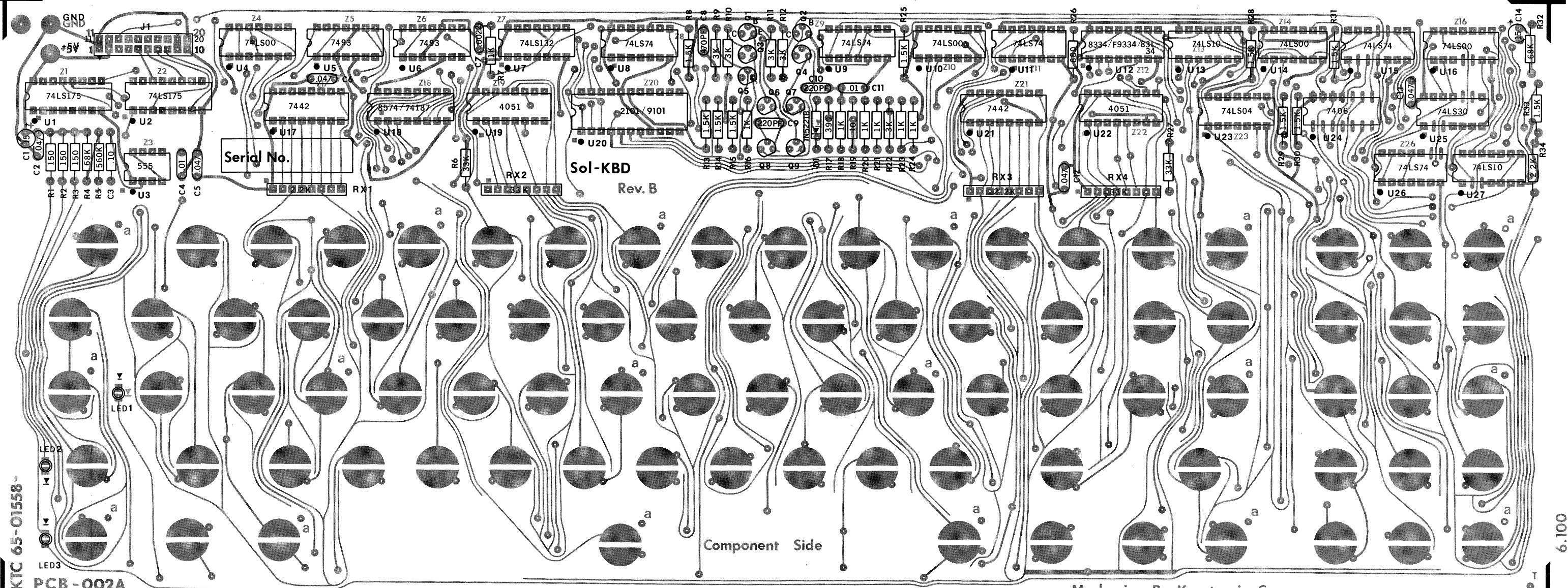
COMP SIDE

PM 2708

ASSEMBLY

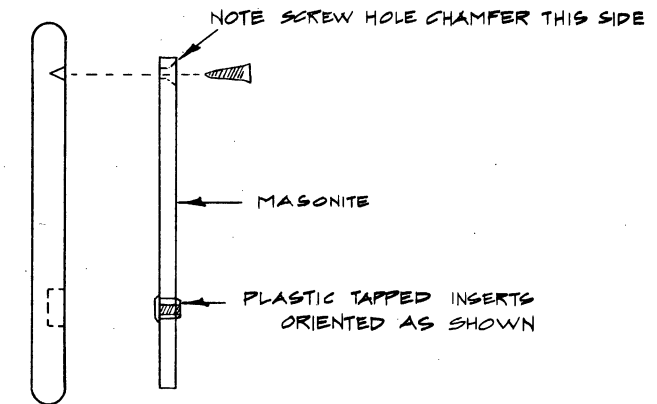
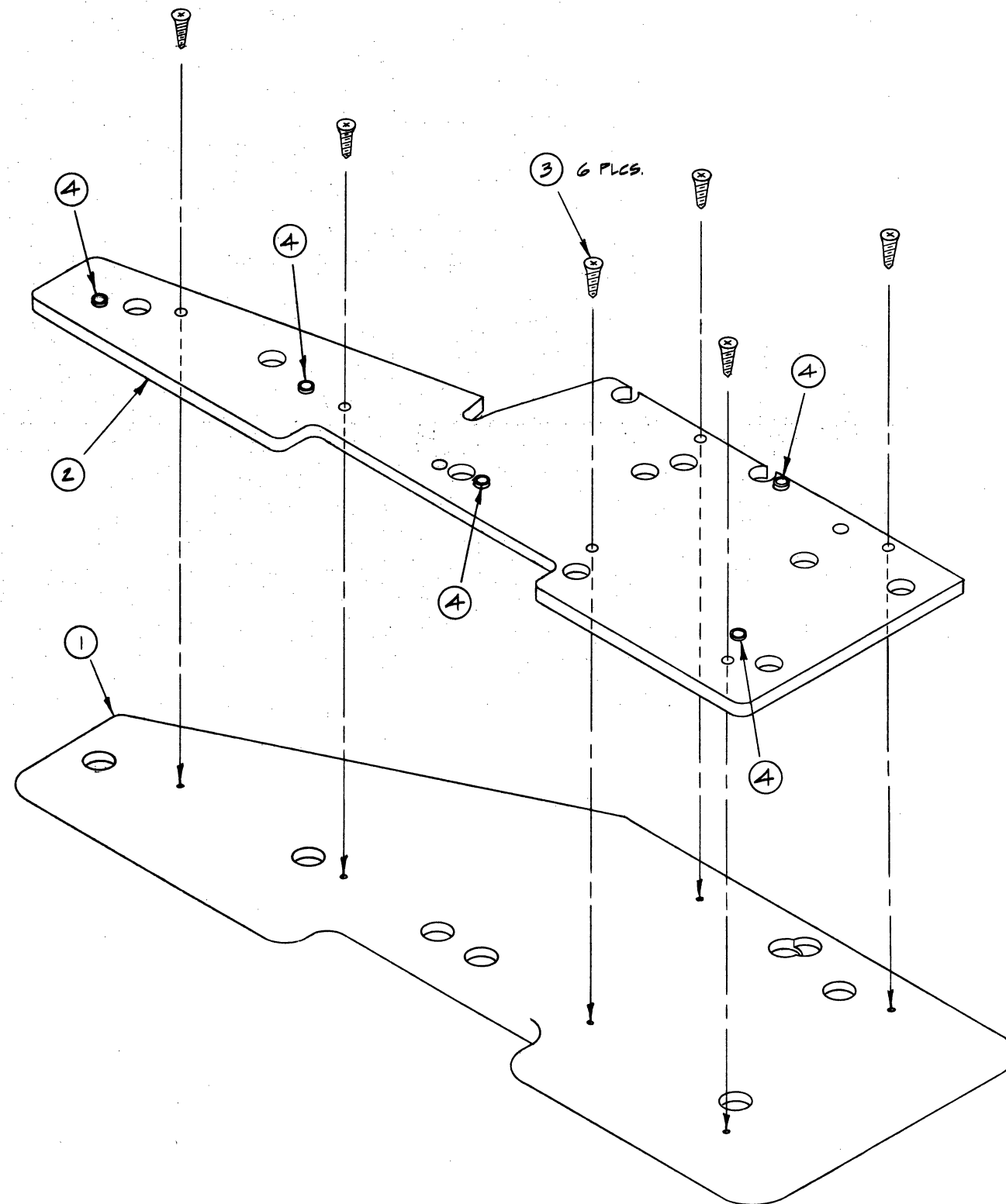
©1977 BY PROCESSOR TECHNOLOGY CORP.  
PM 2708  
MODULE





- NOTES:**
1. LEDs 1, 2, & 3 ARE MV5752
  2. Q1, Q2 & Q9 ARE 2N4274, Q3 - Q8 ARE 2N3640

**Sol-KEYBOARD  
ASSEMBLY**

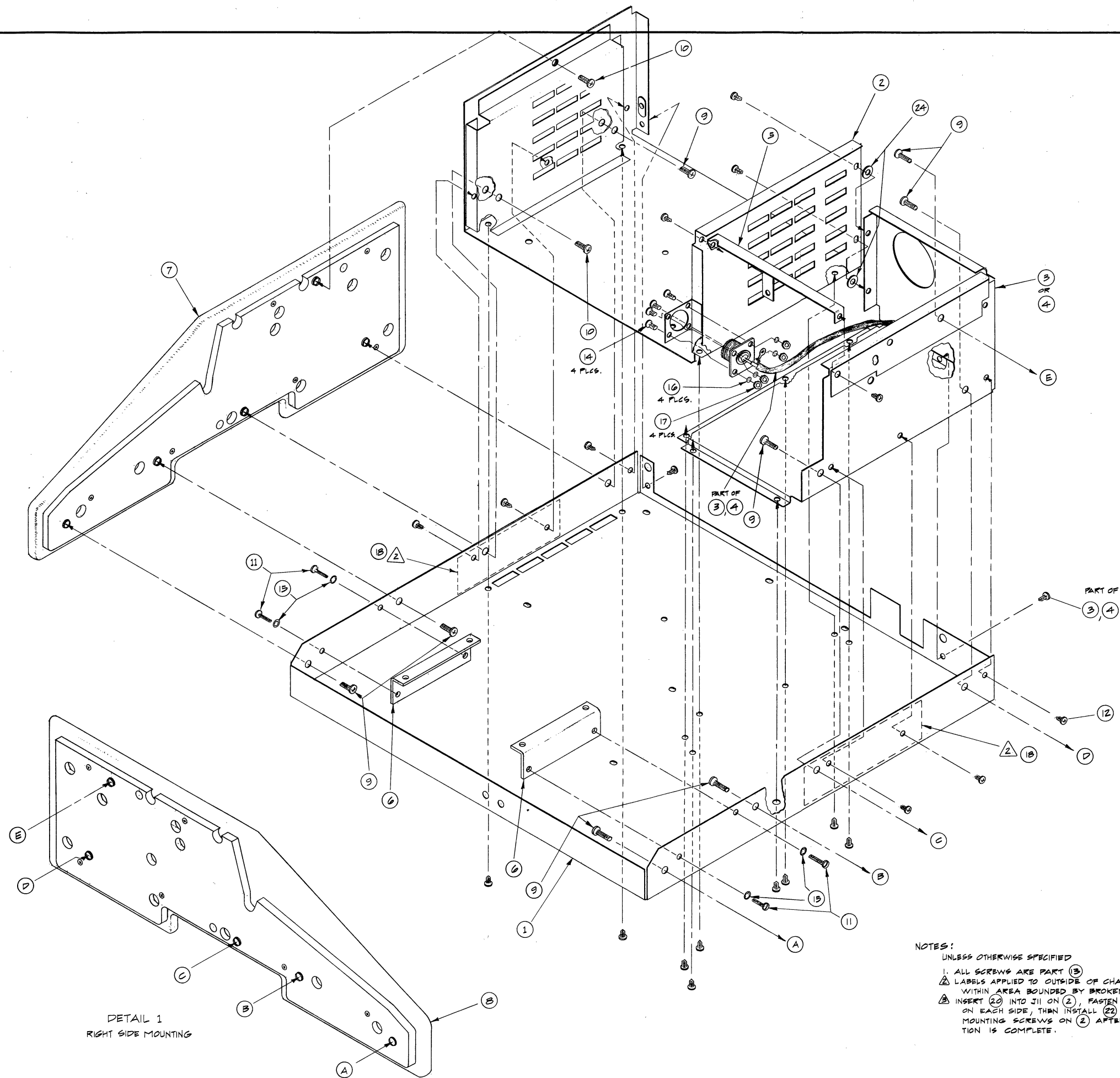


4		NUT, PLASTIC INSERT, G-32	5	
3		SCREW, WOOD, FLAT HEAD #6 x 5/8"	6	
2	101011	SPACER, LEFT SIDE PANEL	1	
1	101009	SLAB, LEFT SIDE PANEL	1	
REF. DES.	PART OR DWG. NO.	PART DESCRIPTION	QTY	

NEXT ASSY.	USED ON	<div><div>D</div><div>Processor Technology Corporation 6200 Hollis Street Emeryville, CA 94608</div></div>	DRAWN: LITO	ENGR:
101000	501		DATE: 1-4-77	DATE:
			CHECKED:	RELEASED:
			DATE:	DATE:
		ASSEMBLY, SIDE PANEL, LEFT HAND		





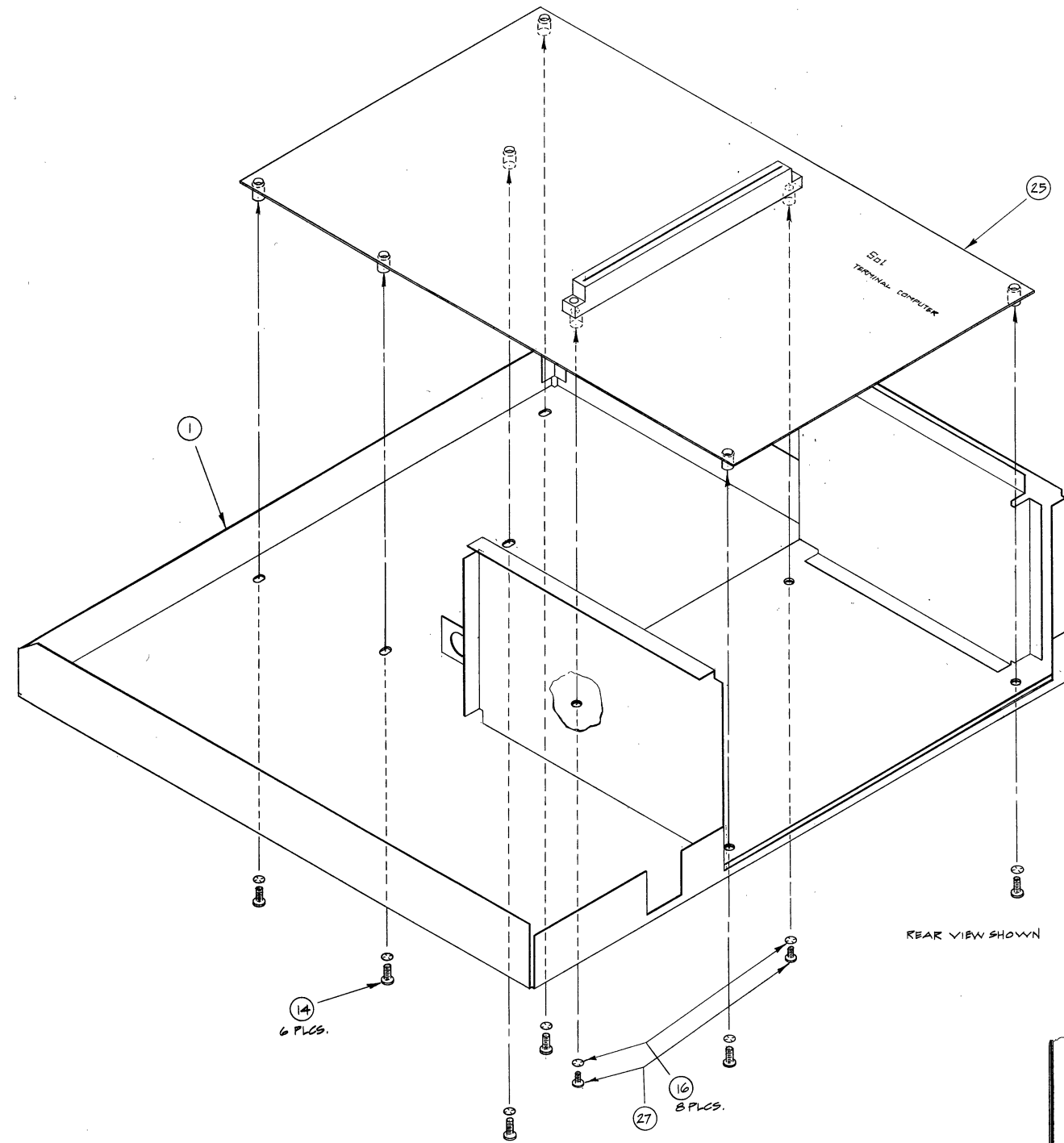


NOTES:  
UNLESS OTHERWISE SPECIFIED  
1. ALL SCREWS ARE PART (13)  
2. LABELS APPLIED TO OUTSIDE OF CHASSIS WITHIN AREA BOUNDED BY BROKEN LINE  
3. INSERT (20) INTO J11 ON (2), FASTEN TO (2) ON EACH SIDE, THEN INSTALL (22) AND (24). TIGHTEN MOUNTING SCREWS ON (2) AFTER INSTALLATION IS COMPLETE.

27		SCREW, MACHINE, 4-40 x 5/16", PAN HEAD	2	2
26	101021	GUSSET, BACKPLANE, LEFT-HAND	0	1
25	102000	SOL PC CIRCUIT CARD	1	1
24		WASHER, FLAT, 1/2" O.D., 3/16" I.D., 1/16" THK	2	2
23		CARD GUIDE	-	10
22	101017	GUSSET, BACKPLANE, RIGHT-HAND	-	1
21	101016	BRACKET, BACKPLANE RIGHT ANGLE	-	2
20	103000	BACK PLANE BOARD ASSEMBLY	-	1
19		SCREW, MACHINE, 4-40 x 5/8", PAN HEAD	-	6
18		LABEL, FINGER WELL, BLACK	2	2
17		NUT, #4-40, HEX	4	4
16		LOCKWASHER, #4, INTERNAL TOOTH	3	3
15		LOCKWASHER, #6, INTERNAL TOOTH	4	4
14		SCREW, MACHINE, 4-40 x 5/16", PAN HEAD	10	16
13		SCREW, SHEET METAL, #6 x 1/2", PAN HEAD	19	31
12		SCREW, SHEET METAL, #6 x 5/16", PAN HEAD	1	1
11		SCREW, MACHINE, 6-32 x 1/2", PAN HEAD	4	4
10		" " " " " " " "	2	2
9		SCREW, MACHINE, 8-32 x 1/2", PAN HEAD	8	8
8	101008	SIDE ASSEMBLY, RIGHT-HAND	1	1
7	101007	SIDE ASSEMBLY, LEFT-HAND	1	1
6	101006	BRACKET, KEYBOARD SUPPORT	2	2
5	101005	BRACKET, CONNECTING	1	1
4	103001-02	POWER SUPPLY, SOL-20	-	1
3	103001-01	POWER SUPPLY, SOL-10	1	-
2	101004	EXPANSION SUBCHASSIS	1	1
1	101003	MAIN CHASSIS	1	1

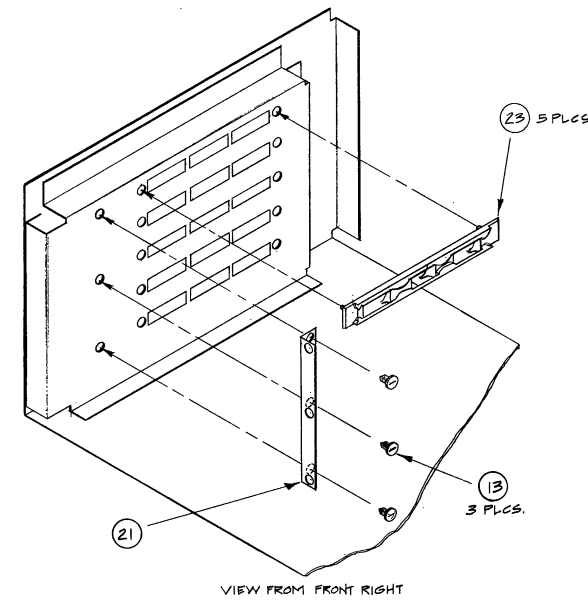
REF. DES.	PART OR DWG. NO.	PART DESCRIPTION	-10-	-20-
-----------	------------------	------------------	------	------

ASSEMBLY, SOL		
SCALE:	APPROVED BY:	DRAWN BY LITO
DATE: 01-17-77	R. MARSH	REVISED
PROCESSOR TECHNOLOGY CORP.		
DRAWING NUMBER		101000

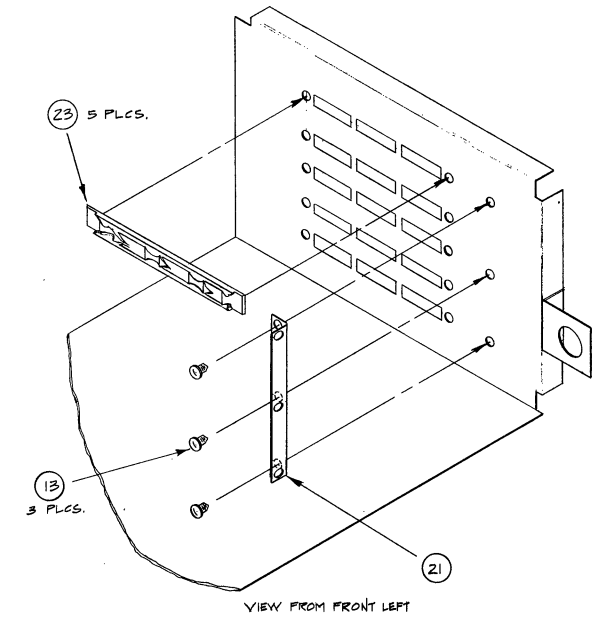


DETAIL - Sol PC INSTALLATION

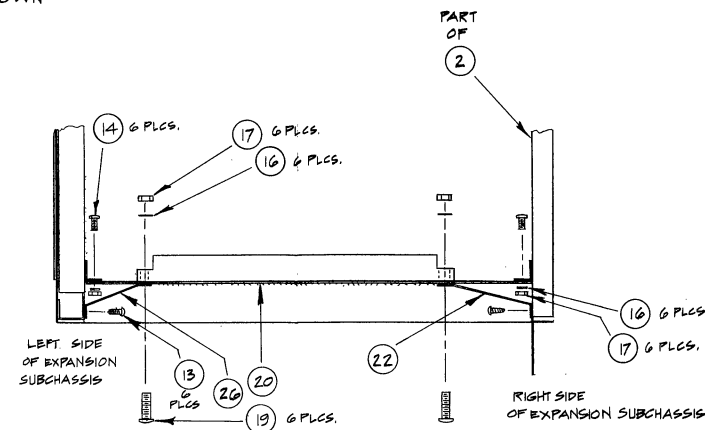
REAR VIEW SHOWN



DETAIL - LEFT SIDE EXPANSION SUBCHASSIS  
BRACKET AND CARD GUIDE MOUNTING.  
(Sol-20 ONLY)

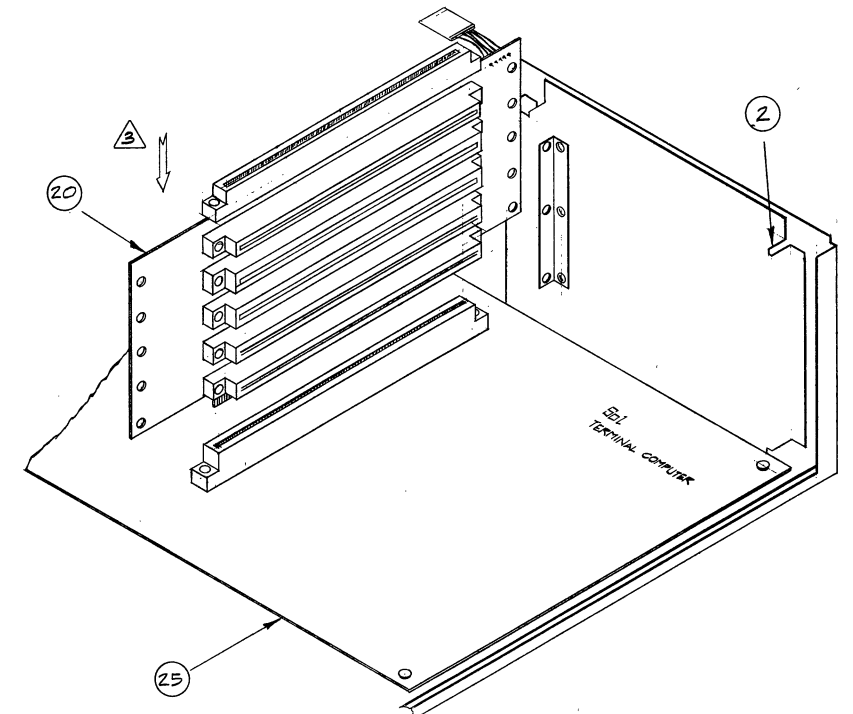


DETAIL - RIGHT SIDE EXPANSION SUBCHASSIS  
BRACKET AND CARD GUIDE MOUNTING.  
(Sol-20 ONLY)



TOP VIEW

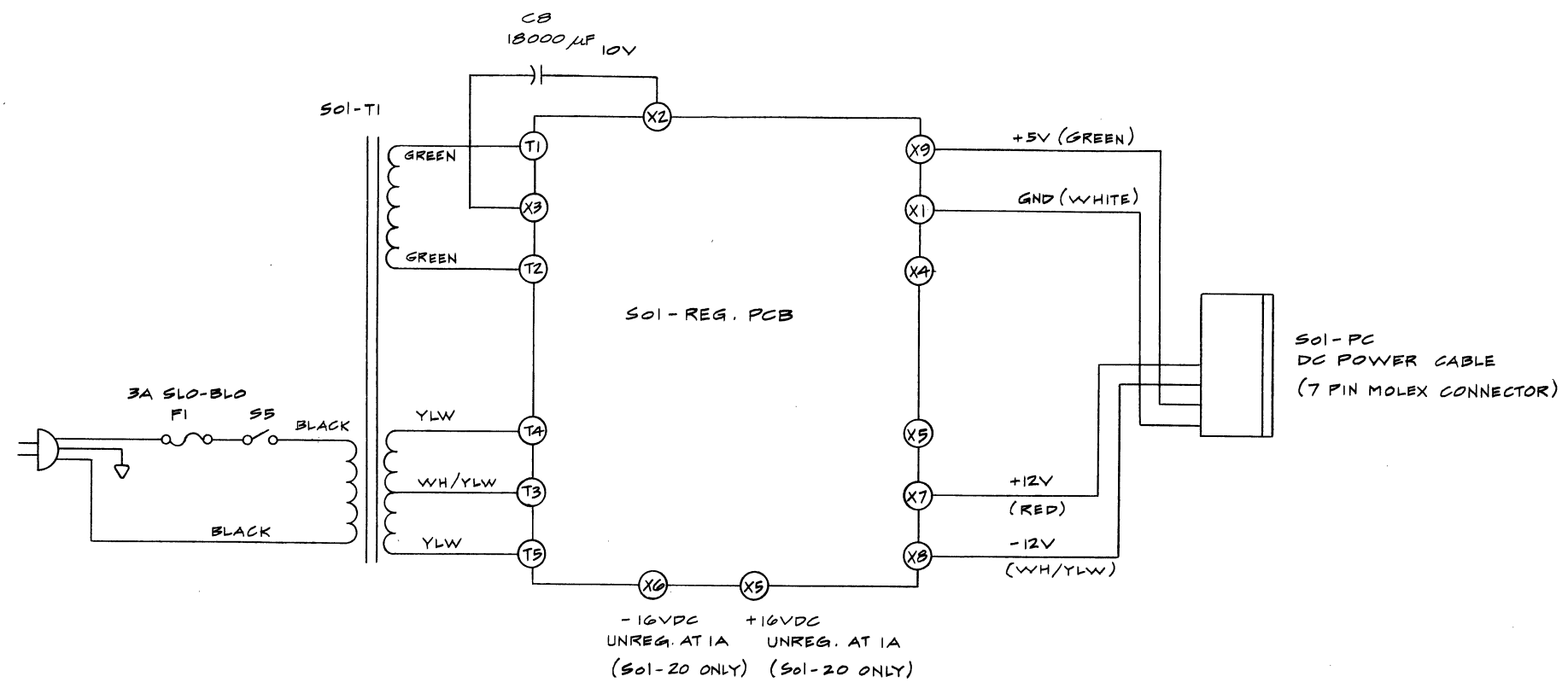
DETAIL - MOUNTING OF BACKPLANE BOARD  
(Sol-20 ONLY)



DETAIL - BACKPLANE BOARD INSERTION INTO J11 OF Sol PC BOARD  
(Sol-20 ONLY)

ASSEMBLY, Sol		
SCALE: —	APPROVED BY: R. MARSH	DRAWN BY LITO
DATE: 01-20-77	REVIEWED	REVIEWED
PROCESSOR TECHNOLOGY CORP.		
REV. A	SHEET 2	DRAWING NUMBER 101000

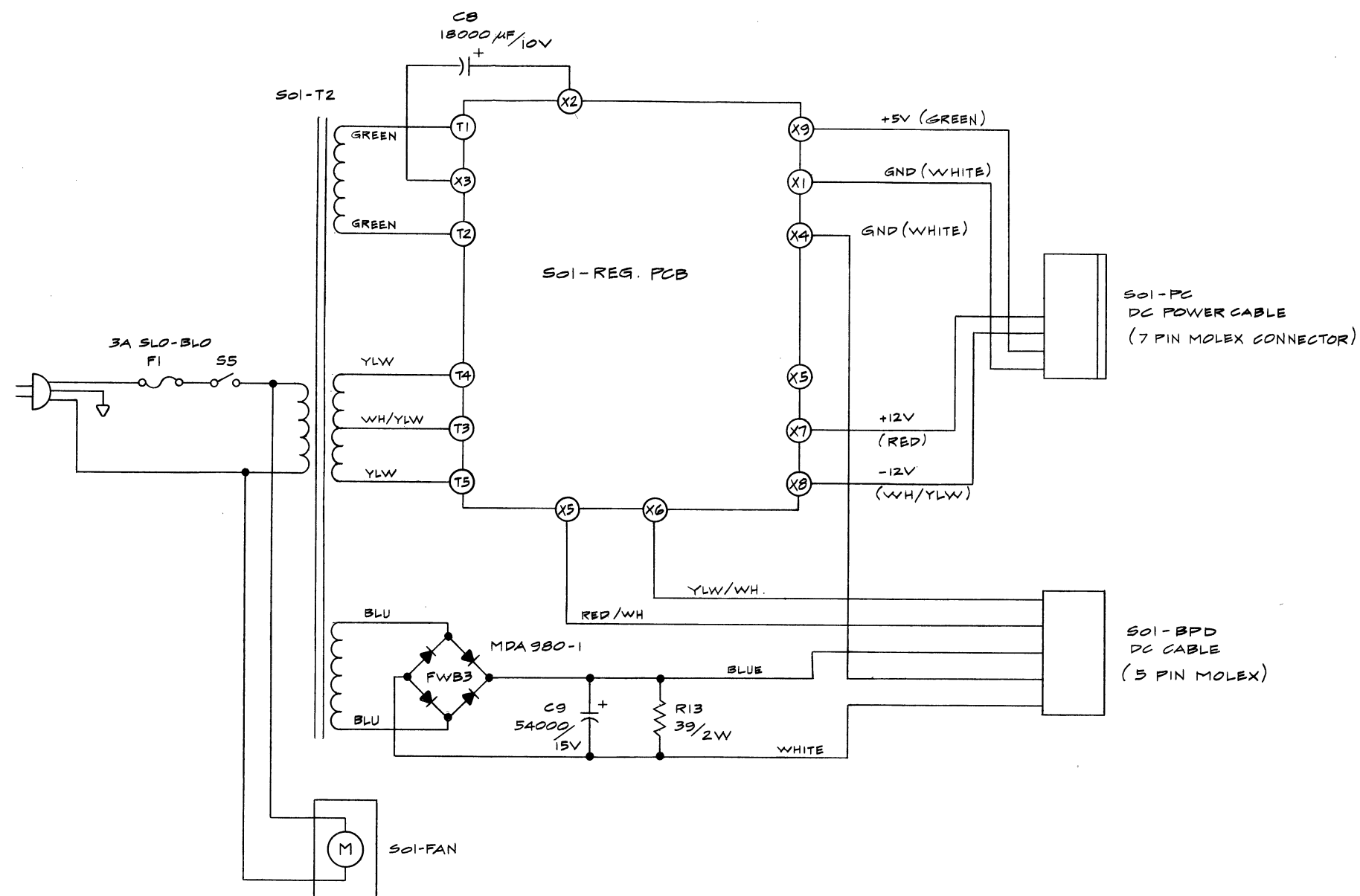




SCHEMATIC, Sol-10 POWER SUPPLY

SCALE: ———	APPROVED BY: R. MARSH	DRAWN BY: LITO
DATE: 11-12-76		

PROCESSOR TECHNOLOGY CORP.



SCHEMATIC, SOI-20 POWER SUPPLY

SCALE: —	APPROVED BY: R. MARSH	DRAWN BY: LIT
DATE: 11-10-76		

PROCESSOR TECHNOLOGY CORP.

Rev. B

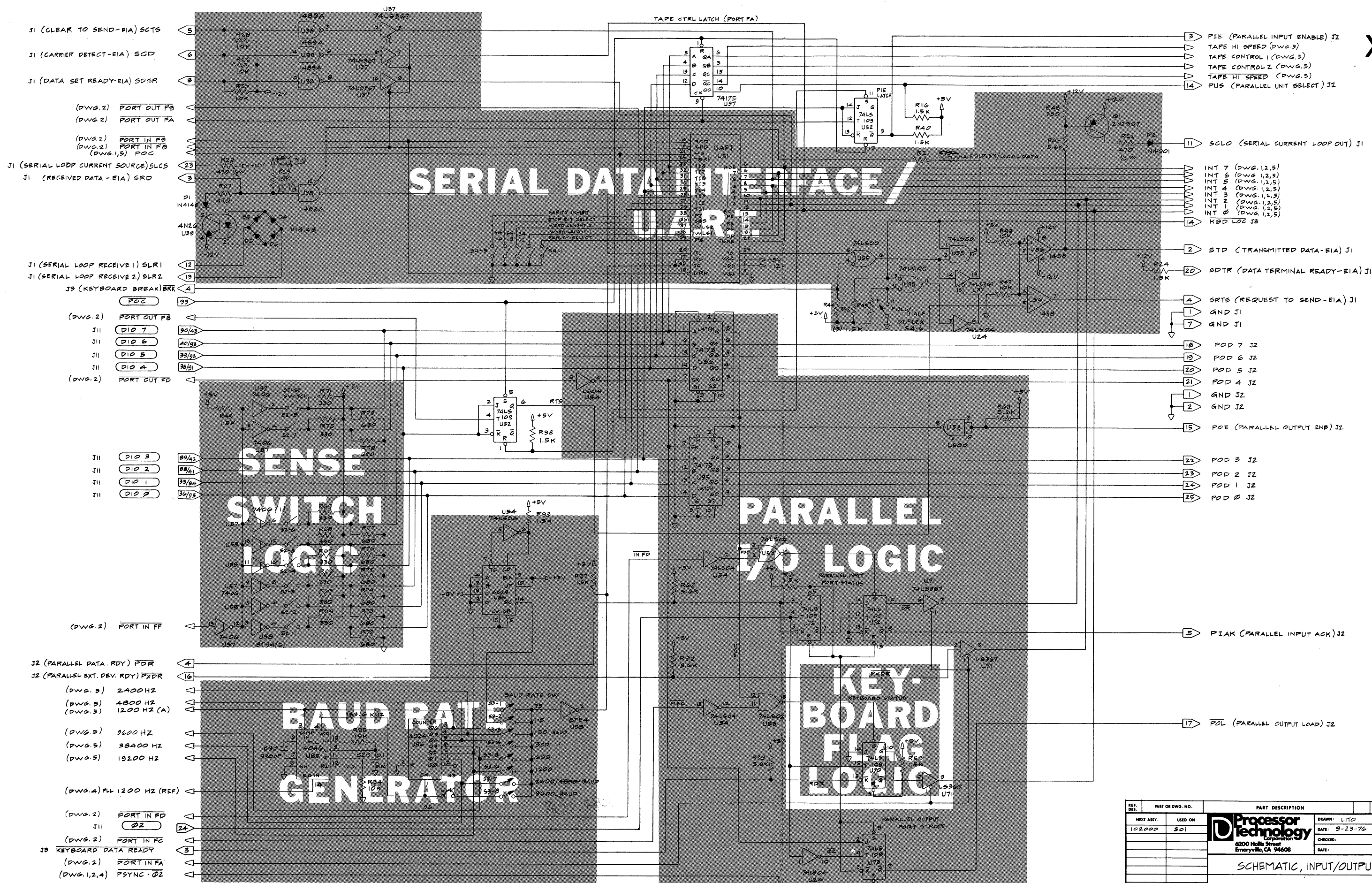









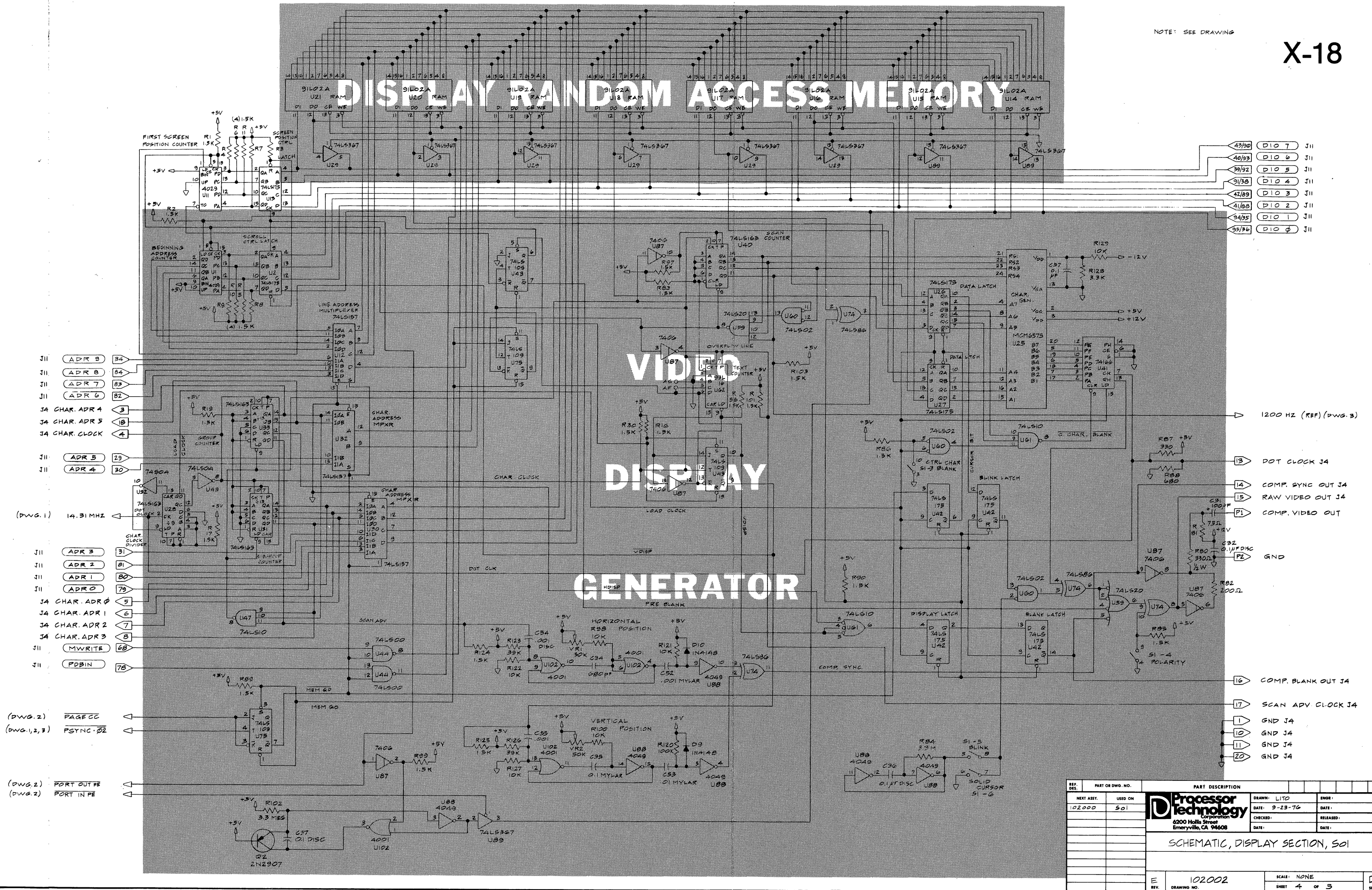




NOTE: SEE DRAWING 5

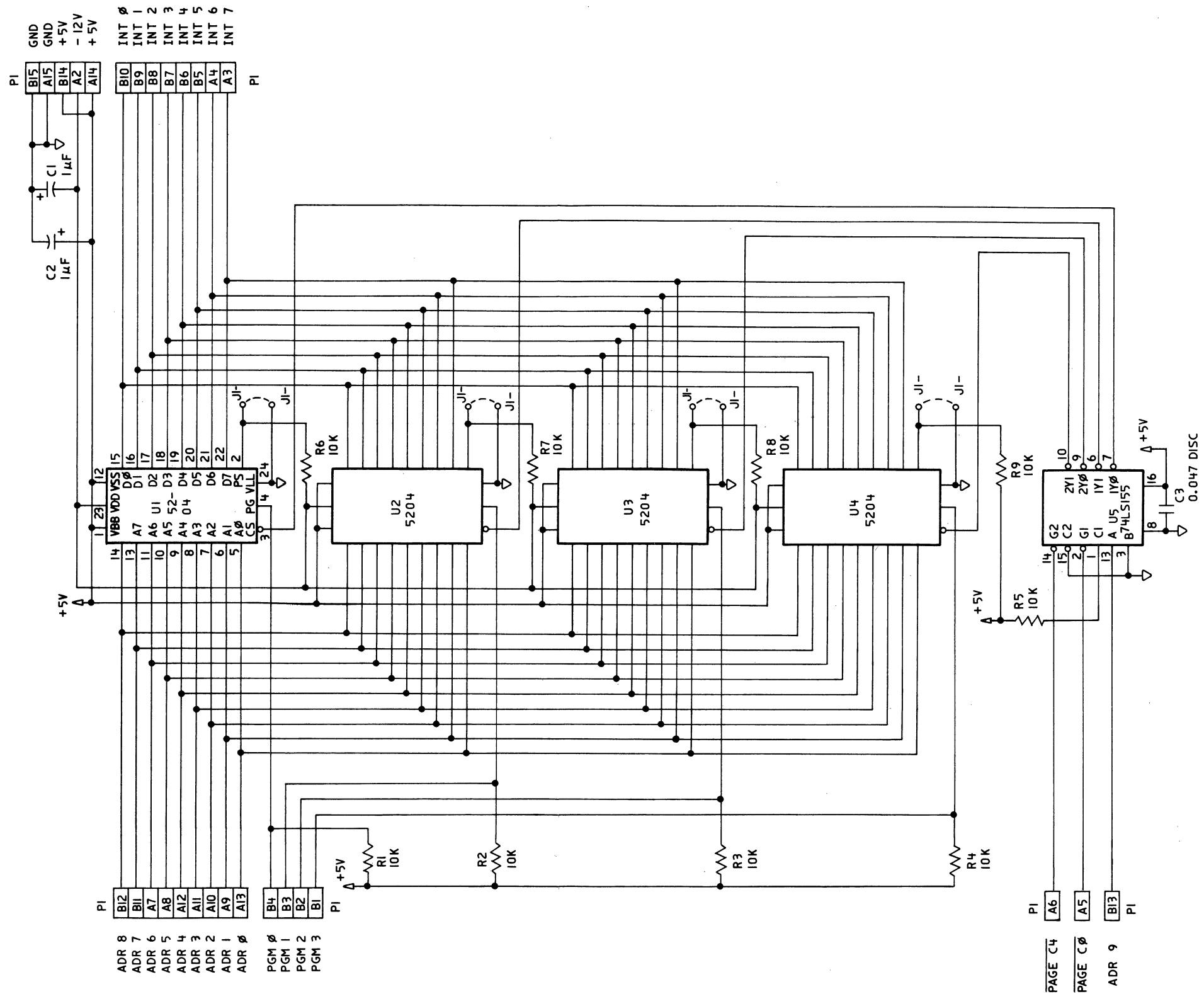
REF. DES.	PART OR DWG. NO.	PART DESCRIPTION							
NEXT ASSY.	USED ON	 <b>Processor Technology</b> Corporation 6200 Hollis Street Emeryville, CA 94608	DRAWN: L110	ENGR:					
102000	501		DATE: 9-23-76	DATE:					
			CHECKED:	RELEASED:					
			DATE:	DATE:					
		SCHEMATIC, INPUT/OUTPUT, Sol							
E REV.	DRAWING NO.	102002	SCALE: NONE						
			SHEET 3	OF 5					



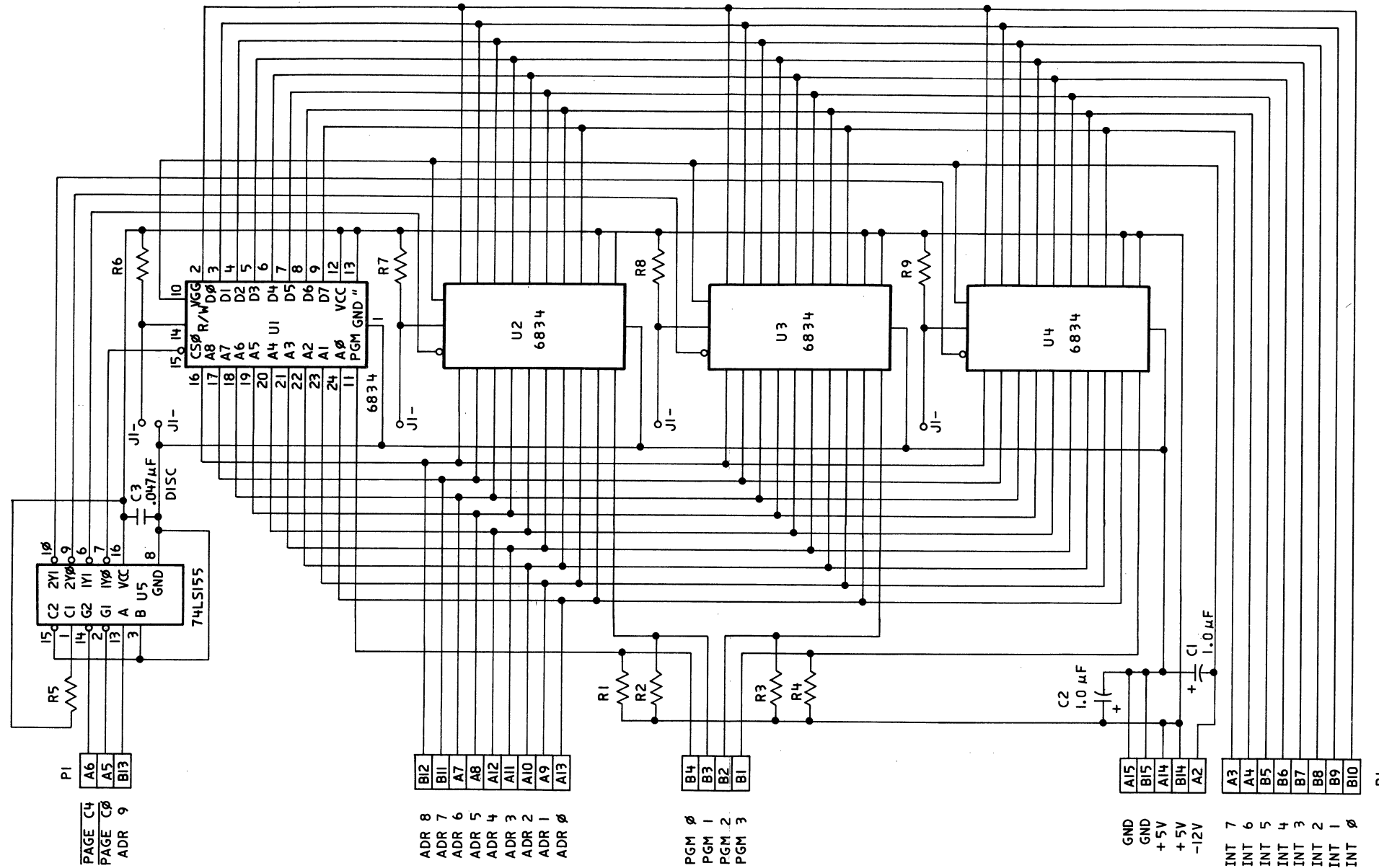






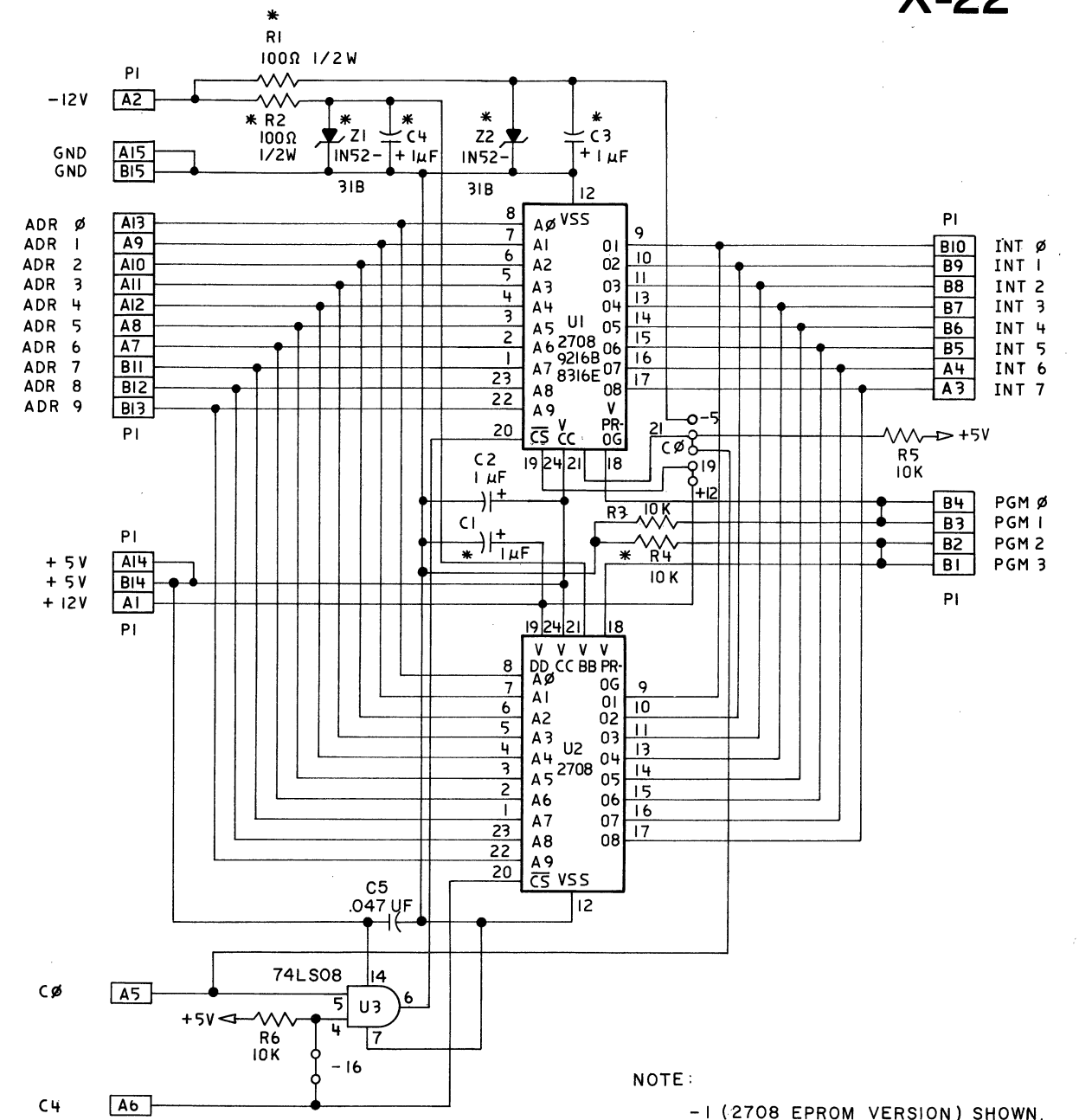


REF. DES.	PART OR DWG. NO.	PART DESCRIPTION	DATE	REV.
NEXT ASSY.	USED ON	108000	501	
		Processor Technology Corporation 6200 Hollis Street Emeryville, CA 94608	DRAWN: LITO DATE: 10-19-76	ENG: DATE:
			CHECKED: DATE:	RELEASED: DATE:
SCHEMATIC, PERSONALITY MODULE 5204				
REV. 108002		SCALE: NONE		C
DRAWING NO.		SHEET 1 OF 1		SIZE



NOTE: ALL RESISTORS 10K, 1/4W, 5%

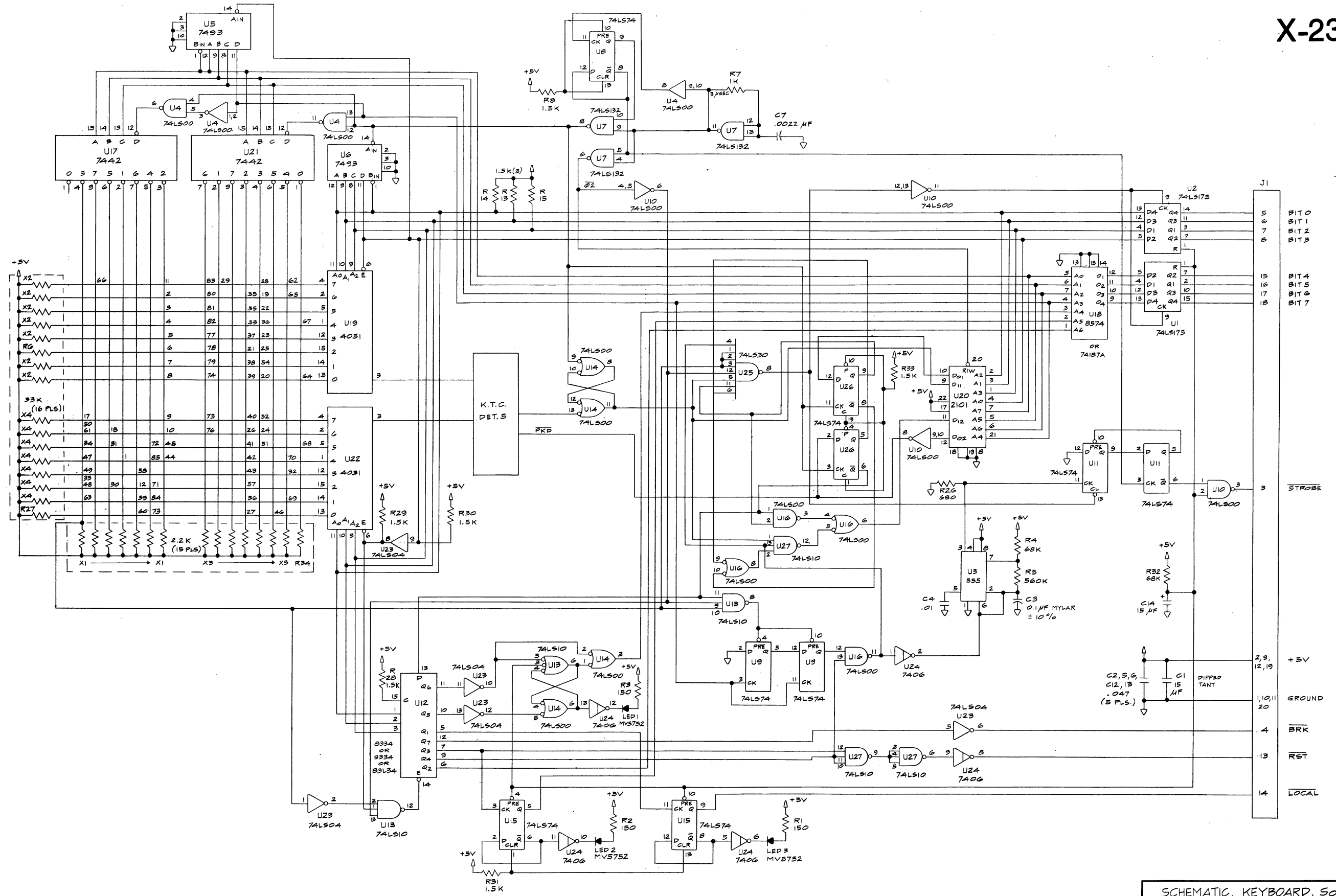
SCHEMATIC, Personality Module 6834		
SCALE: _____	APPROVED BY: ROBERT M. MARSH	DRAWN BY: LITO
DATE: 10-20-76	CHECKED BY: LEE	
PROCESSOR TECHNOLOGY CORP.		
Sol		



NOTE:  
-1 (2708 EPROM VERSION) SHOWN.  
\* = PART WHICH IS MISSING FROM  
REV. 0 (9216 MASKED ROM  
VERSION)

REF. DES.	PART OR DWG. NO.	PART DESCRIPTION	DRAWN: LITO	ENGR:
NEXT ASSY.	USED ON	<b>D Processor Technology Corporation</b> 6200 Hollis Street Emeryville, CA 94608	DATE: 10-20-76	DATE:
107000	501		CHECKED:	RELEASED:
			DATE:	DATE:
SCHEMATIC, PERSONALITY MODULE 2708				
REV. B	DRAWING NO. 07002	SCALE: NONE	B SIZE	



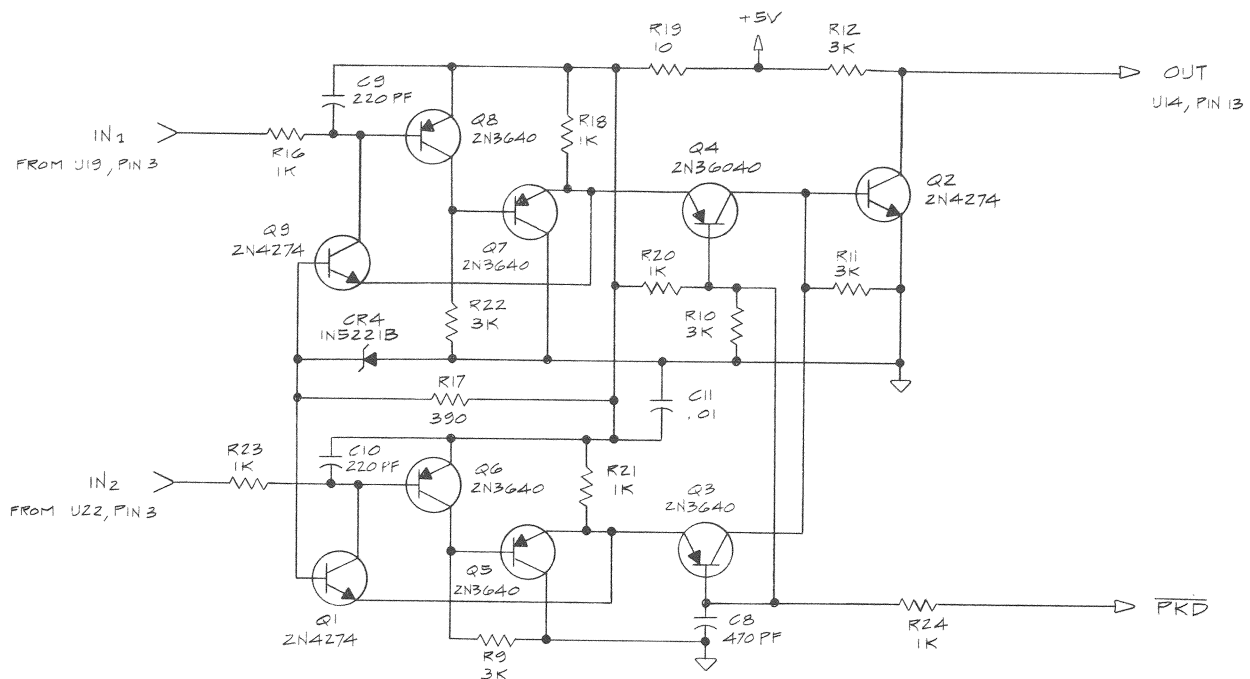




## Sol MANUAL ADDENDUM #1

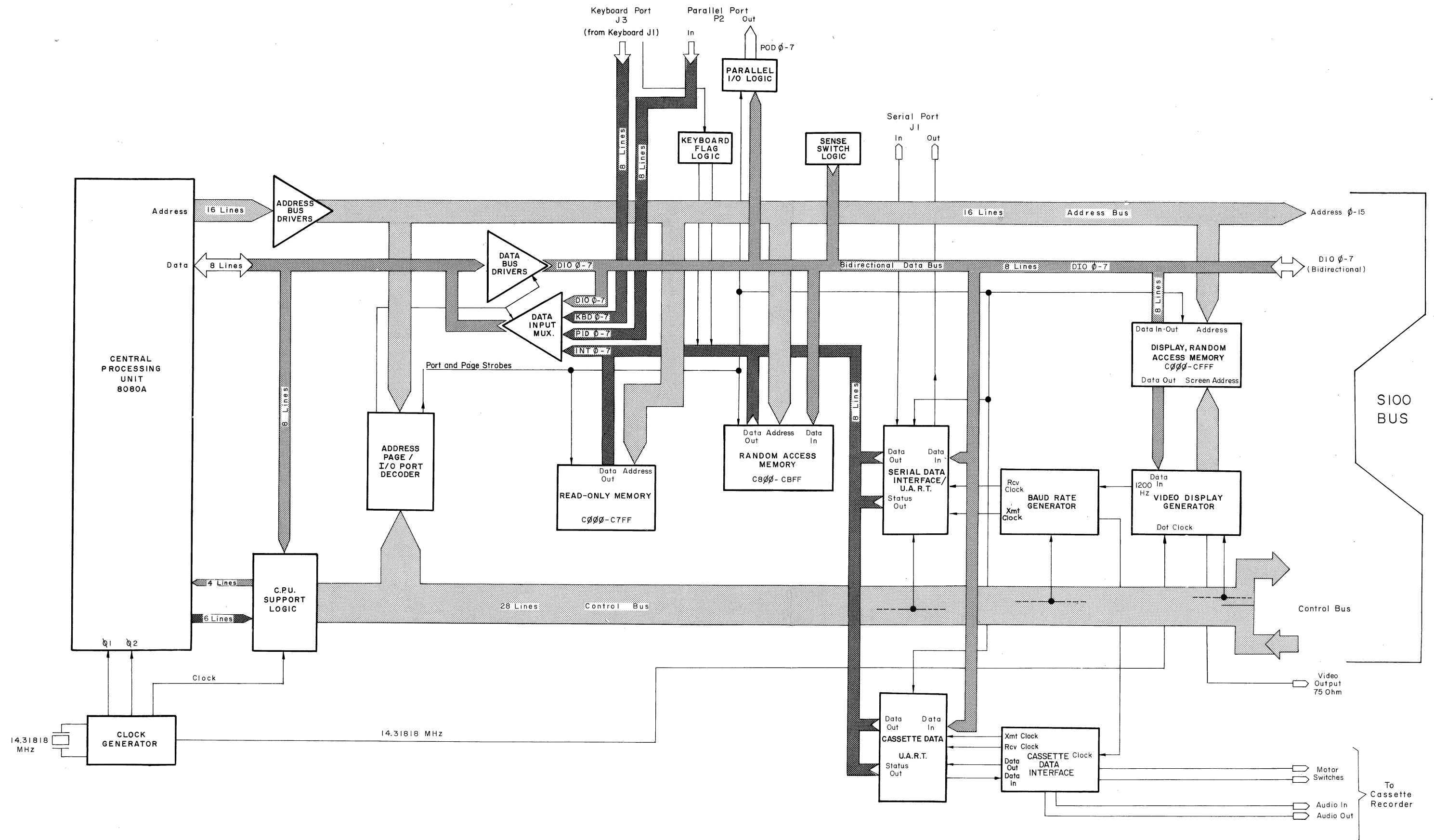
Reference Section X, Drawing X-23.

A block function labelled "K.T.C." is shown between U-19 and U22, and U14. This block contains the Capacitive Switch Detector Circuit. The parts constituting this circuit are listed, and the assembly covered in Section V of this manual. The theory of operation is covered in Section VIII. At the time of publication of this manual, operation of this circuit was proprietary information, but has now been released. The schematic is shown below. Note on the schematic X-23 that this detail is shown here on this page.



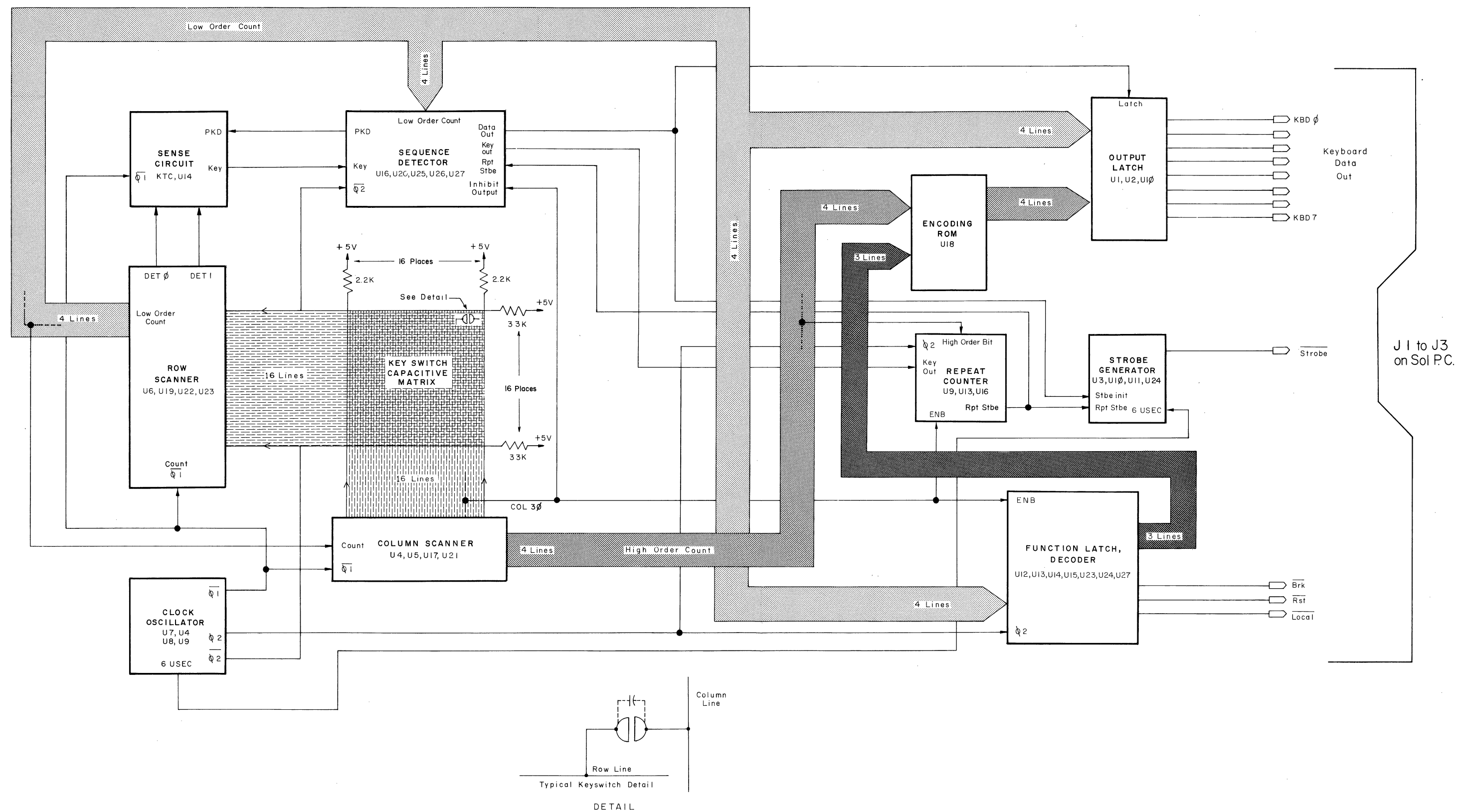
# Sol-PC. BLOCK DIAGRAM-Rev. D,E

X-24



BLOCK FUNCTIONS (in alphabetic order)	SCHEMATIC PAGE	USES IC NO's
1. ADDRESSES BUS DRIVERS	X15	U67,68,81
2. ADDRESS PAGE/I/O PORT DECODER	X15,X16	U22,23,24,34,35, 36,44,47,48,53, 61,83
3. BAUD RATE GENERATOR	X17	U84,85,86
4. CASSETTE DATA INTERFACE	X19	U99,100,101,108, 109,110,113,U69, U98,U110,U111,U112
5. CASSETTE DATA U.A.R.T.	X19	U69
6. CENTRAL PROCESSING UNIT	X15	U105
7. CLOCK GENERATOR	X15	U77,90,91,92,104
8. C.P.U. SUPPORT LOGIC	X15	U44,45,46,48,49, 50,76,77,93,106,107
9. DATA BUS DRIVERS	X15	U80,81
10. DATA INPUT MULTIPLEXER	X15	U65,66,77,78
11. DISPLAY RANDOM ACCESS MEMORY	X18	U14-21,U29,U89,U44, U75
12. KEYBOARD FLAG LOGIC	X17	U53,54,70,71
13. PARALLEL I/O LOGIC	X17	U53,54,71,72,73, 95,96
14. RANDOM ACCESS MEMORY (RAM)	X16	U3,4,5,6,7,8,9, 10,24
15. READ-ONLY MEMORY (ROM)	X16	(SEE SEC. IV)
16. SENSE SWITCH LOGIC	X17	U57,58
17. SERIAL DATA INTERFACE U.A.R.T.	X17	U37,38,39,24,51, 55,56
18. VIDEO DISPLAY GENERATOR	X18	U1,2,11,12,13,25, 26,27,28,30,31,32, 33,40,41,42,43,47, 59,60,61,62,74,75, 87,88,89,U92,102

**X-25**

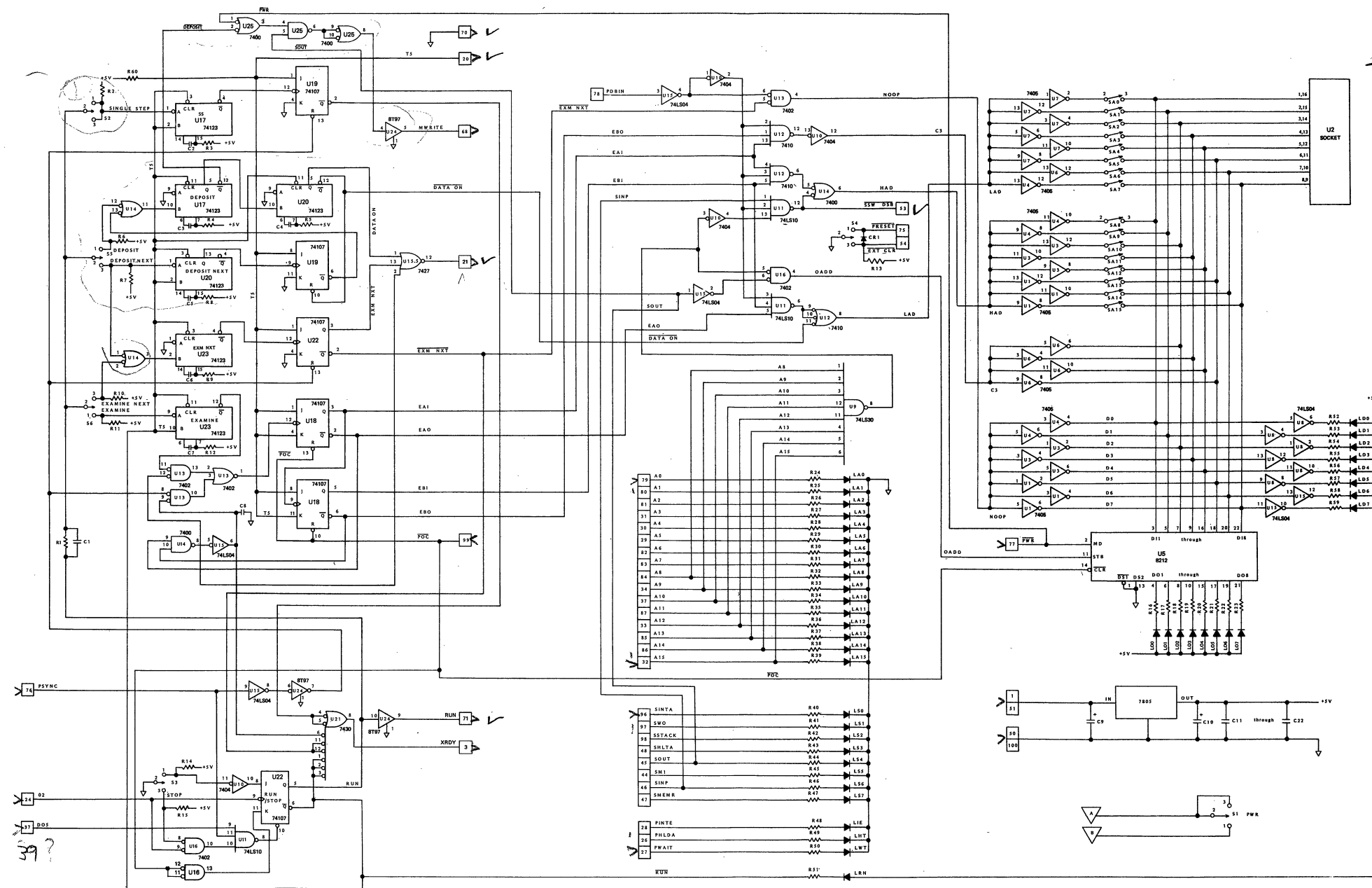




SoI - KEYBOARD  
PHOTO

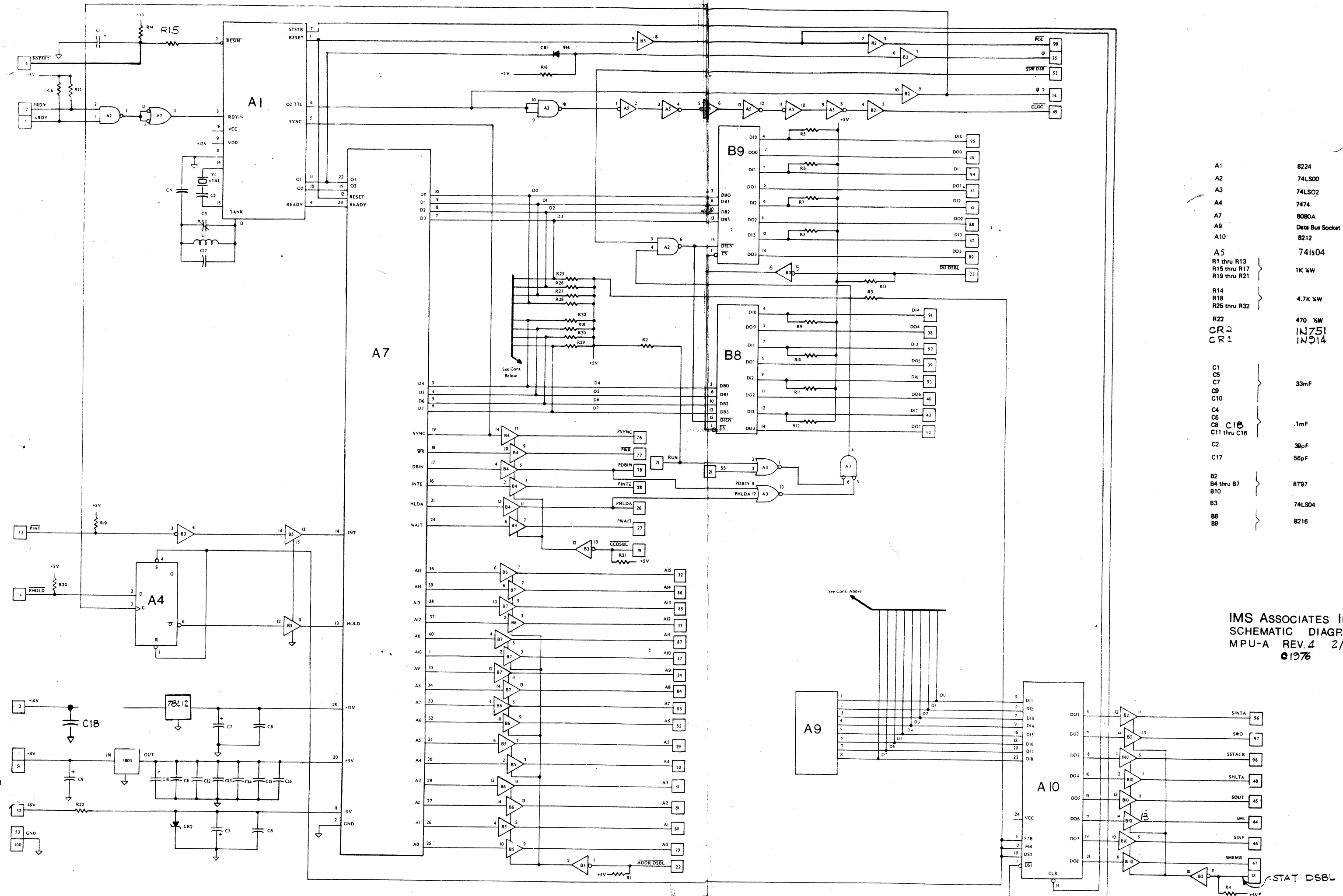
MPV-A

REVISIONS			
LTR	DESCRIPTION	DATE	APPROVED
0	ORIGINAL	12/75	
1	ECN76-0051 & ECO77-0039	11/76 & 4/77	
2	ECO77-0098	8/77	



U1		LO0	
U3		THRU	
U4	7405	LO7	
U6		LA0	
U7		THRU	
		LA15	
U2	DATA	LD0	
	HEADER	THRU	
	SOCKET	LD7	LED'S
	8212	LS0	
U5		THRU	
U8	74LS04	LS7	
U9	74LS30	LIE	
U10	7404	LHD	
U11	74LS10	LRN	
U12	7410	LWT	
U13			
U16	7402	R1	270K 1/4W
U14		R2	
U25	7400	R6	
U15.5	7427	R7	
U17		R10	
U20	74123	R11	1K 1/4W
U23		R13	
U18		THRU	
U19	74107	R15	
U22		R60	
U21	7430	R3	
U24	8T97	R4	
		R5	47K 1/4W
C1		R8	
C2		R9	
C5		R12	
C6	.1 uF	R16	220 1/4W
C7		THRU	
C11		R59	
C22			
C3	.001 uF		
C8			
C4	.01 uF		
C9			
C10	33 uF		
CR1	1N914		

TOLERANCES UNLESS OTHERWISE SPECIFIED FRACTIONS DEC. ANGLES ± ± ±		© 1977 IMSAI MFG. CORP., SAN LEANDRO, CA. ALL RIGHTS RESERVED WORLDWIDE MADE IN U.S.A.		
APPROVALS		IMSAI 8080 SYSTEM CPA REV. 4 SCHEMATIC DIAGRAM		
DATE				
DRAWN				
CHECKED		SCALE	SIZE B	DRAWING NO.
		DO NOT SCALE DRAWING		
		SHEET		



- |              |                 |
|--------------|-----------------|
| A1           | 8224            |
| A2           | 74LS00          |
| A3           | 74LS02          |
| A4           | 7474            |
| A7           | 8080A           |
| A9           | Data Bus Socket |
| A10          | 8212            |
| A5           | 74LS04          |
| R1 thru R13  | 1K 1/4W         |
| R15 thru R17 |                 |
| R19 thru R21 |                 |
| R14          | 4.7K 1/4W       |
| R18          |                 |
| R25 thru R32 |                 |
| R22          | 470 1/4W        |
| CR2          | 1N751           |
| CR1          | 1N514           |
| C1           | 33mF            |
| C5           |                 |
| C7           |                 |
| C9           |                 |
| C10          | .1mF            |
| C4           |                 |
| C6           |                 |
| C8           |                 |
| C11 thru C16 | 56pF            |
| C2           |                 |
| C17          |                 |
| B2           | 8T97            |
| B4 thru B7   |                 |
| B10          |                 |
| B3           | 74LS04          |
| B8           | 8216            |
| B9           |                 |

IMS ASSOCIATES INC.  
SCHEMATIC DIAGRAM  
MPU-A REV. 4 2/76  
©1976



## APPENDICES

- AI Statement of Warranty
- AII 8080 Operating Codes
- AIII Standard Color Code
- AIV Loading DIP Devices, Soldering Tips,  
and Installing Augat Pins
- AV IC Pin Configurations
- AVI TV Interface
- AVII Pin-outs for Connectors  
S100 Bus Definitions, Switch Functions,  
and Bit Assignments
- AVIII "Your Personal Genie",  
(an article on types of software)

LOADING DIP (DUAL IN-LINE PACKAGE) DEVICES

Most DIP devices have their leads spread so that they can not be dropped straight into the board. They must be "walked in" using the following procedure:

- (1) Orient the device properly. Pin 1 is indicated by a small embossed dot on the top surface of the device at one corner. Pins are numbered counterclockwise from pin 1.
- (2) Insert the pins on one side of the device into their holes on the printed circuit card. Do not press the pins all the way in, but stop when they are just starting to emerge from the opposite side of the card.
- (3) Exert a sideways pressure on the pins at the other side of the device by pressing against them where they are still wide below the bend. Bring this row of pins into alignment with its holes in the printed circuit card and insert them an equal distance, until they begin to emerge.
- (4) Press the device straight down until it seats on the points where the pins widen.
- (5) Turn the card over and select two pins at opposite corners of the device. Using a fingernail or a pair of long-nose pliers, push these pins outwards until they are bent at a 45° angle to the surface of the card. This will secure the device until it is soldered.

SOLDERING TIPS

- (1) Use a low-wattage iron--25 watts is good. Larger irons run the risk of burning the printed-circuit board. Don't try to use a soldering gun, they are too hot.
- (2) Use a small pointed tip and keep it clean. Keep a damp piece of sponge by the iron and wipe the tip on it after each use.
- (3) Use 60-40 rosin-core solder ONLY. DO NOT use acid-core solder or externally applied fluxes. Use the smallest diameter solder you can get.

NOTE: DO NOT press the top of the iron on the pad or trace. This will cause the trace to "lift" off of the board which will result in permanent damage.

- (4) In soldering, wipe the tip, apply a light coating of new solder to it, and apply the tip to both parts of the joint, that is, both the component lead and the printed-circuit pad. Apply the solder against the lead and pad being heated, but not directly to the tip of the iron. Thus, when the solder

melts the rest of the joint will be hot enough for the solder to "take", (i.e., form a capillary film).

- (5) Apply solder for a second or two, then remove the solder and keep the iron tip on the joint. The rosin will bubble out. Allow about three or four bubbles, but don't keep the tip applied for more than ten seconds.
- (6) Solder should follow the contours of the original joint. A blob or lump may well be a solder bridge, where enough solder has been built upon one conductor to overflow and "take" on the adjacent conductor. Due to capillary action, these solder bridges look very neat, but they are a constant source of trouble when boards of a high trace density are being soldered. Inspect each integrated circuit and component after soldering for bridges.
- (7) To remove solder bridges, it is best to use a vacuum "solder puller" if one is available. If not, the bridge can be reheated with the iron and the excess solder "pulled" with the tip along the printed circuit traces until the lump of solder becomes thin enough to break the bridge. Braid-type solder remover, which causes the solder to "wick up" away from the joint when applied to melted solder, may also be used.

#### INSTALLING AUGAT PINS

Augat pins are normally supplied on carriers (e.g., 8-pin and 16-pin carriers). In many cases the PC board layout permits Augat pins to be installed while still attached to the carrier or a portion of the carrier. In other cases the pins must be installed singly.

To install two or more pins that are still attached to the carrier, proceed as follows:

#### NOTE

It is perfectly alright to appropriately cut a carrier to accommodate the installation. For example, an 8-pin carrier can be cut in half (4 pins each) across the short dimension to fit a 4-pin, 4-corner layout. It may also be cut in half along the long dimension to fit a 4-pin, inline layout.

- (1) Insert pins in the mounting holes from the front (component) side of board. (The carrier will hold the pins perpendicular to the board.)
- (2) Solder all pins from back (solder) side of board so the solder "wicks up" to the front side.



- (3) Check for solder bridges.
- (4) Remove carrier.

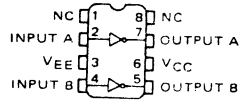
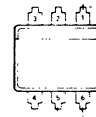
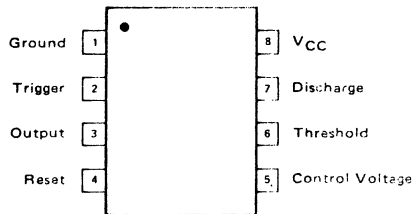
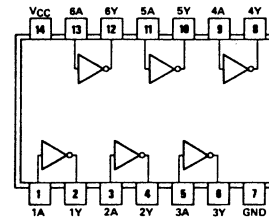
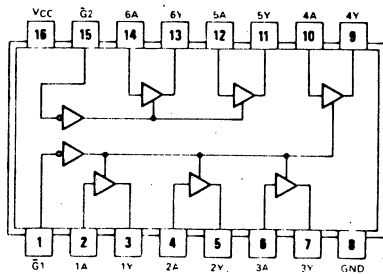
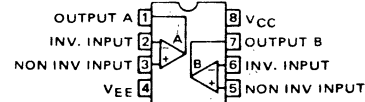
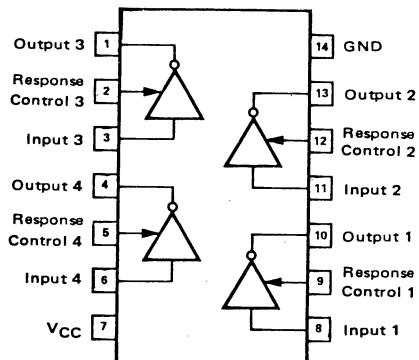
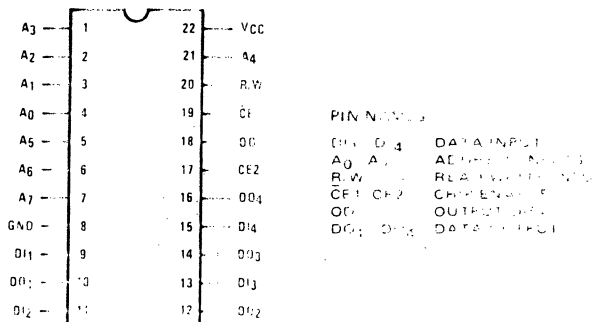
To install single pins, proceed as follows:

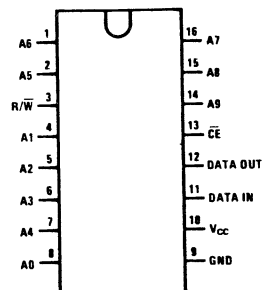
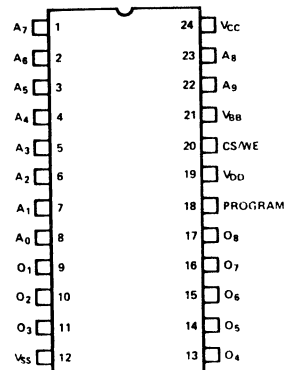
- (1) Hold board between two objects so that it stands on edge.
- (2) Insert pins in the mounting holes from front (component) side of board.
- (3) Solder pins from back (solder) side of board so the solder "wicks up" to the front side. (This will hold the pins firmly in place.)
- (4) Insert a component lead into one pin and reheat the solder. Using the component lead, adjust pin until it is perpendicular to board. Allow solder to cool while holding the pin as steady as possible. Remove component lead. Repeat this procedure with other pins.

NOTE

If cooled solder is mottled or crystallized, a "cold joint" is indicated, and the solder should be reheated.

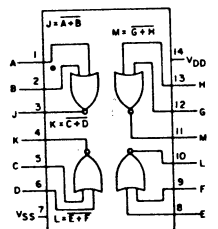
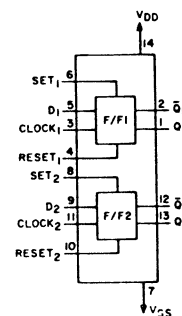
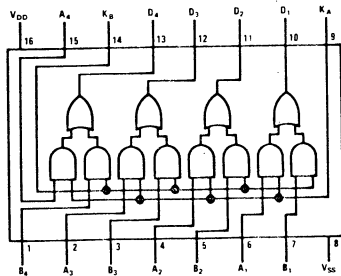
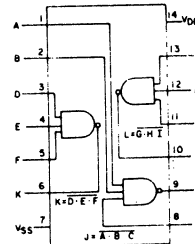
- (5) Check each installation for cold joints and solder bridges.

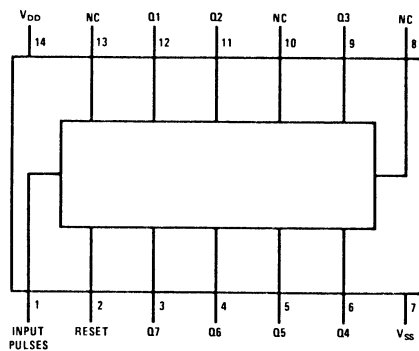
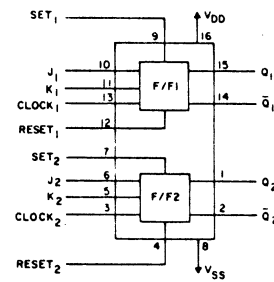
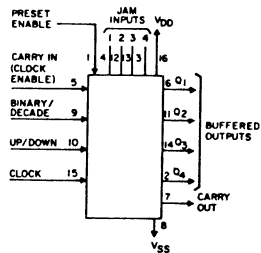
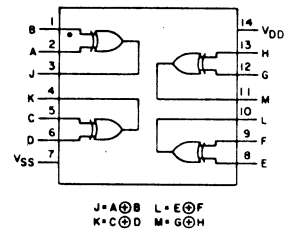
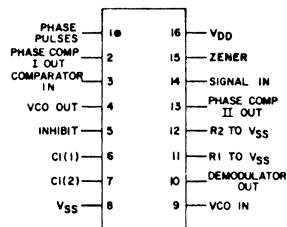
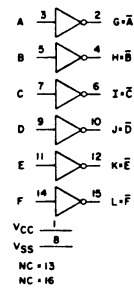
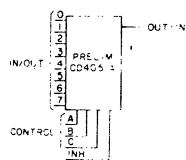
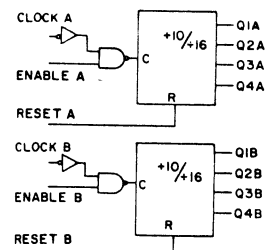
00264N265558T948T97 or 74367145814892101

21L02 or 91L022708

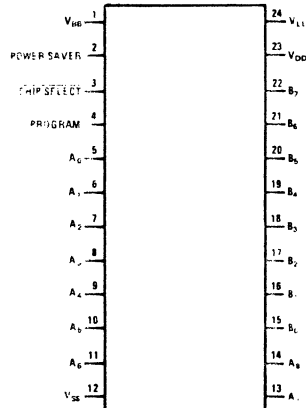
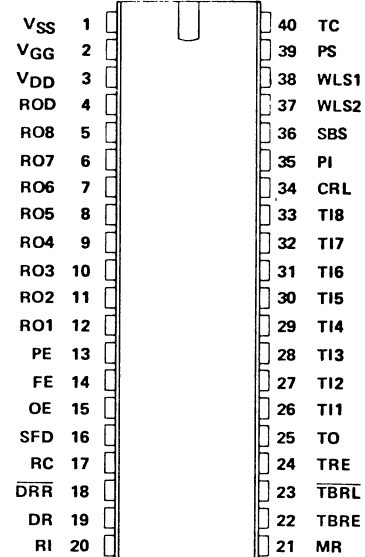
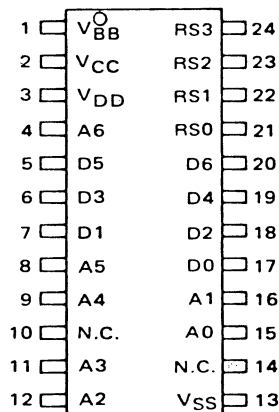
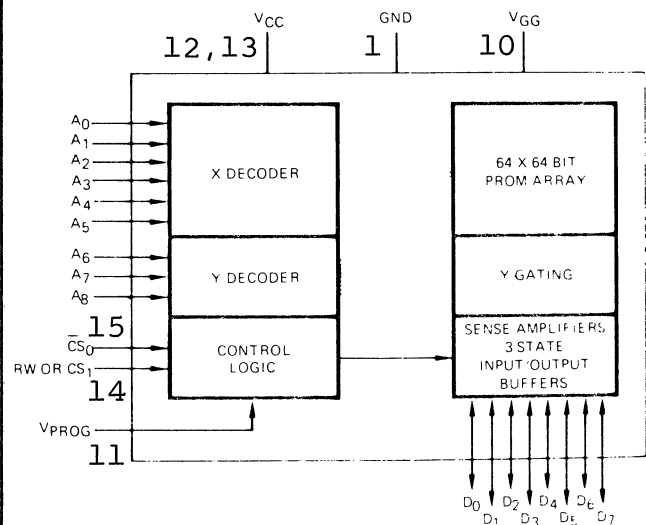
## PIN NAMES

A <sub>0</sub> -A <sub>9</sub>	ADDRESS INPUTS
O <sub>1</sub> -O <sub>8</sub>	DATA OUTPUTS
CS	CHIP SELECT INPUTS

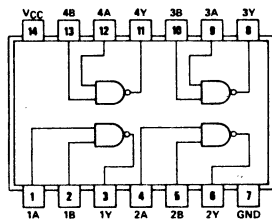
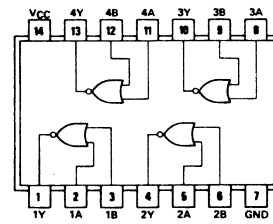
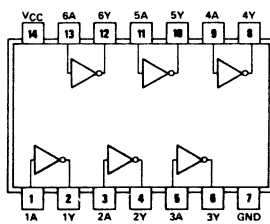
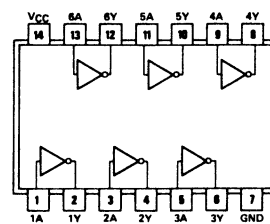
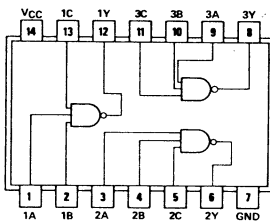
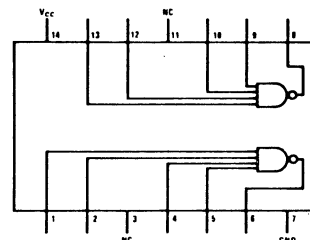
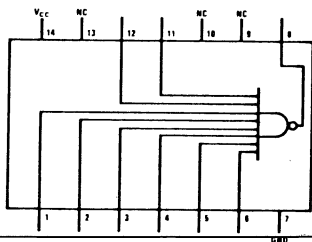
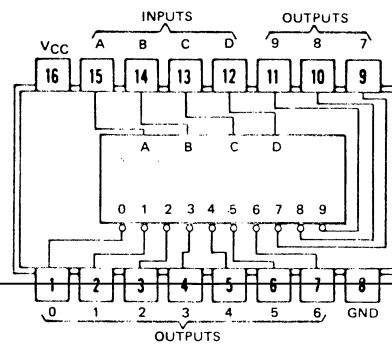
4001401340194023

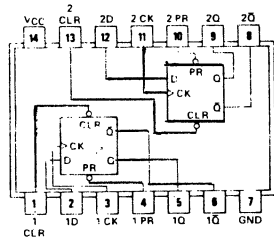
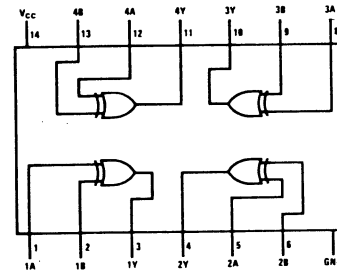
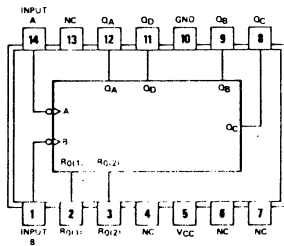
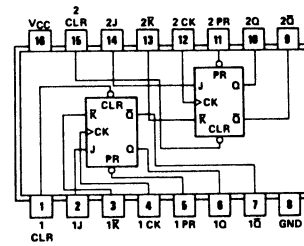
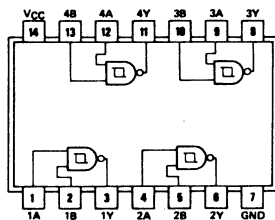
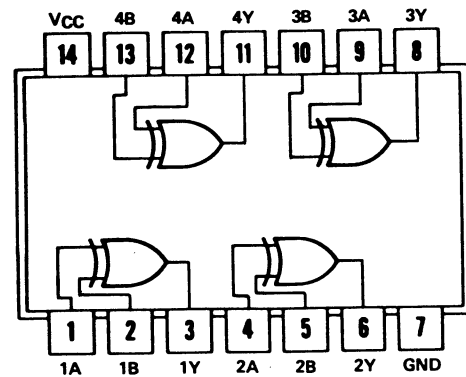
40244027402940304046404940514520

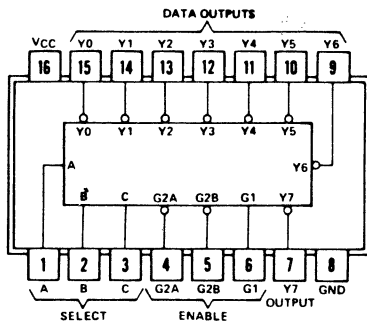
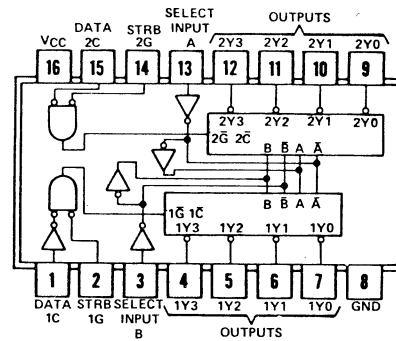
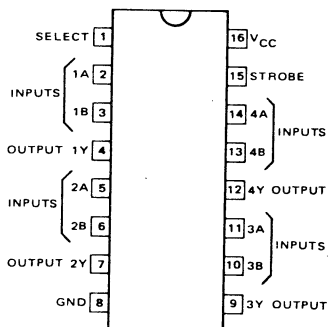
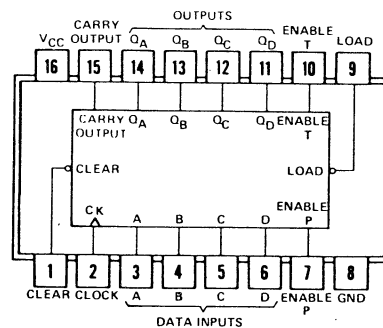


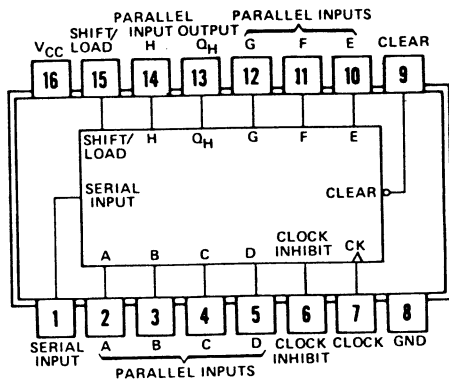
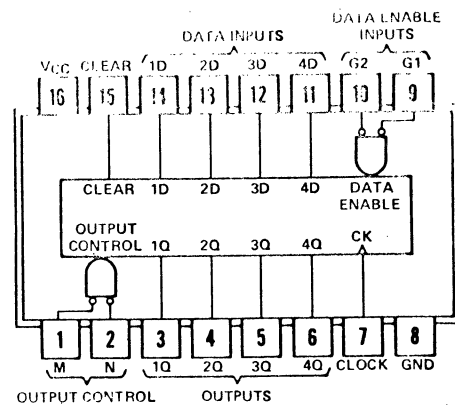
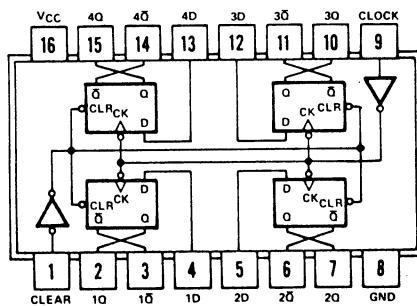
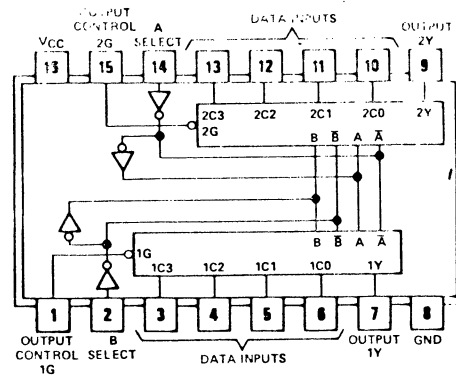
520460116574 or 65756834

A0 - 24	A5 - 19	D1 - 3	D6 - 8
A1 - 23	A6 - 18	D2 - 4	D7 - 9
A2 - 22	A7 - 17	D3 - 5	
A3 - 21	A8 - 16	D4 - 6	
A4 - 20	D0 - 2	D5 - 7	

74007402740474067410742074307442

747474867493741097413274136

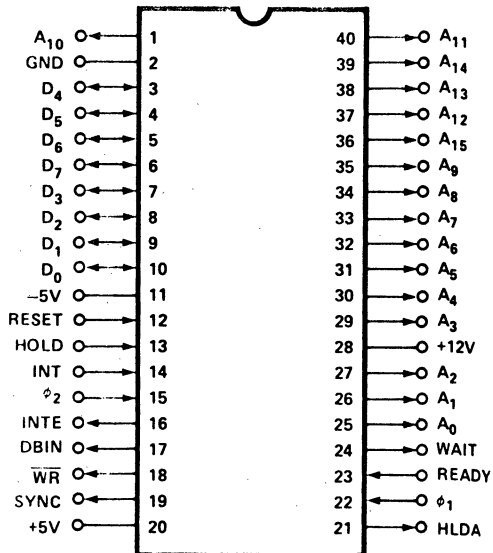
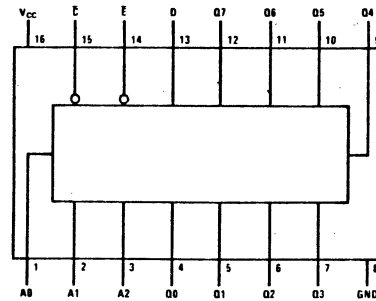
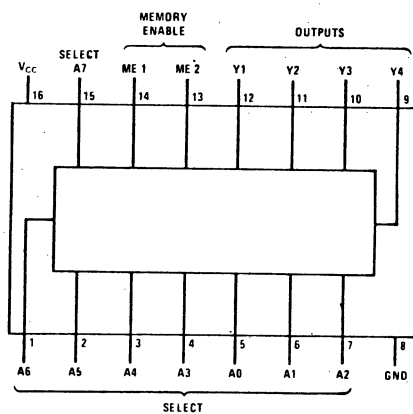
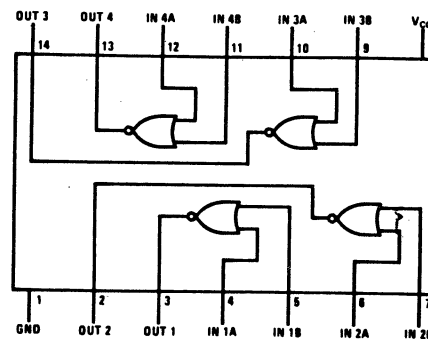
74138741557415774163 or 93L16

741667417374175742537812 and 7912

Pin 1 Input (Base)  
Pin 2 Output (Emitter)  
Pin 3 Ground (Collector)

Heat sink surface connected  
to pin 3



8080833485748836

# Television Interface

Anyone with a bunch of memory circuits, control logic and a wire wrap gun can whip up a digital video generator with TTL output levels. The problem as I see it is to get that digital video signal into a form that the TV set can digest. The care and feeding of digital inputs to the TV set is the subject of Don Lancaster's contribution to *BYTE 2* — an excerpt from his forthcoming book, *TV Typewriter Cookbook*, to be published by Howard W. Sams, Indianapolis, Indiana.

... CARL

We can get between a TV typewriter and a television style display system either by an rf modulator or a direct video method.

In the rf modulator method, we build a miniature, low power, direct wired TV transmitter that clips onto the antenna terminals of the TV set. This has the big advantage of letting you use any old TV set and ending up with an essentially free display that can be used just about anywhere. No set modifications are needed, and you have the additional advantage of automatic safety isolation and freedom from hot chassis shock problems.

There are two major restrictions to the rf modulator method. The first of these is that transmitters of this type must meet

certain exactly spelled out FCC regulations and that system type approval is required. The second limitation is one of bandwidth. The best you can possibly hope for is 3.5 MHz for black and white and only 3 MHz for color, and many economy sets will provide far less. Thus, long character line lengths, sharp characters, and premium (lots of dots) character generators simply aren't compatible with clip-on rf entry.

In the direct video method, we enter the TV set immediately following its video detector but before sync is picked off. A few premium TV sets and all monitors already have a video input directly available, but these are still expensive and rare. Thus, you usually have to modify your TV set, either

adding a video input and a selector switch or else dedicating the set to exclusive TV typewriter use. Direct video eliminates the bandwidth restrictions provided by the tuner, i-f strip, and video detector filter. Response can be further extended by removing or shorting the 4.5 MHz sound trap and by other modifications to provide us with longer line lengths and premium characters. No FCC approval is needed, and several sets or monitors are easily driven at once without complicated distribution problems.

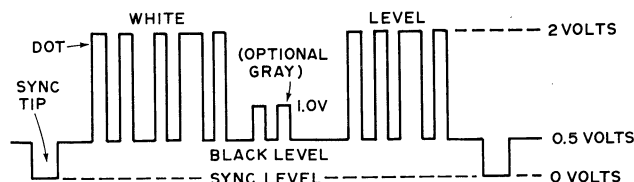
There are two limitations to the direct video technique. One is that the set has to be modified to provide direct video entry. A second, and far more severe, restriction, is that many television sets are "hot chassis" or ac-dc sets with one side of their chassis connected to the power line. These sets introduce a severe shock hazard and cannot be used as TV typewriter video entry displays unless some isolation technique is used with them. If the TV set has a power transformer, there is usually no hot chassis problem. Transistor television sets and IC sets using no vacuum tubes tend to have power transformers, as do older premium tube type sets. All others (around half the sets around today) do not.

## Direct Video Methods

With either interface approach, we usually start by getting the dot matrix data, blanking, cursor, and sync signals together into one composite video signal whose

by  
Don Lancaster  
Box 1112  
Parker AZ 85344

Fig. 1. Standard video interface levels. (Source impedance = 72 or 100 Ohms.)





form is useful to monitors and TV sets. A good set of standards is shown in Fig. 1. The signal is dc coupled and always positive going. Sync tips are grounded and blacker than black. The normal open circuit black level is positive by one-half a volt, and the white level is two volts positive. In most TV camera systems, intermediate levels between the half volt black level and the two volt white level will be some shade of gray, proportionately brighter with increasing positive voltage. With most TV typewriter systems, only the three states of zero volts (sync), half a volt (black), and two volts (white dot) would be used. One possible exception would be an additional one volt dot level for a dim but still visible portion of a message or a single word.

The usual video source impedance is either 72 or 100 Ohms. Regardless of how far we travel with a composite video output, some sort of shielding is absolutely essential.

For short runs from board to board or inside equipment, tightly twisted conductors should be OK, as should properly guarded PC runs. Fully shielded cables should be used for interconnections between the TVT and the monitor or TV set, along with other long runs. As long as the total cable capacitance is less than 500 pF or so (this is around 18 feet of RG178-U

miniature coax), the receiving end of the cable need not be terminated in a 72 or 100 Ohm resistor. When terminated cable systems are in use for long line runs or multiple outputs, they should be arranged to deliver the signal levels of Fig. 1 at their output under termination. Generally, terminated cable systems should be avoided as they need extra in the way of drivers and supply power.

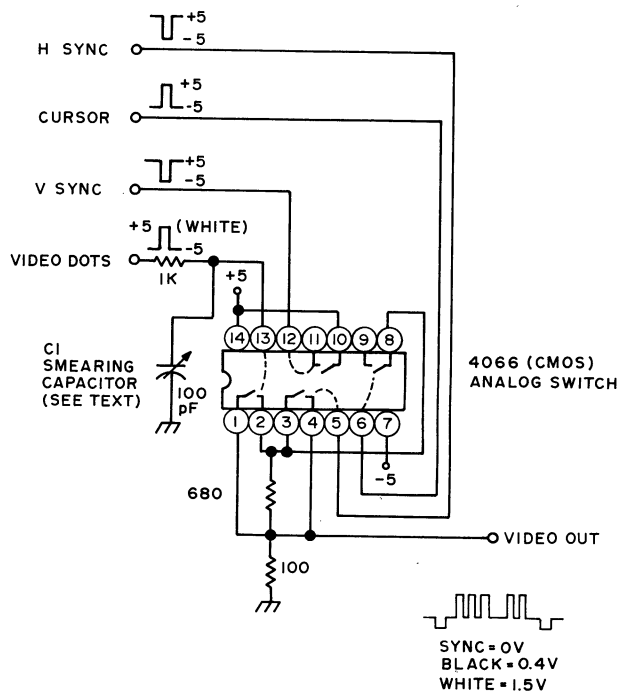
The exact width of the horizontal and vertical sync pulses isn't usually too important, so long as the shape and risetime of these pulses are independent of position control settings and power supply variations. One exception to this is when you're using a color receiver and a color display. Here, the horizontal sync pulse should be held closely to 5.1 microseconds, so the receiver's color burst sampling does in fact intercept a valid color burst. More on this later.

#### Intentional Smear

Fig. 2 shows us a typical composite video driver using a 4066 quad analog switch. It gives us a 100 Ohm output impedance and the proper signal levels. Capacitor C1 is used to purposely reduce the video rise and fall times. It is called a smearing capacitor.

Why would we want to further reduce the bandwidth and response of a TV system that's already hurting to begin with? In the case of a quality video monitor, we wouldn't. But if we're using an ordinary run-of-the-mill TV set, particularly one using rf entry, this capacitor can

Fig. 2. Analog switch combiner generates composite video.



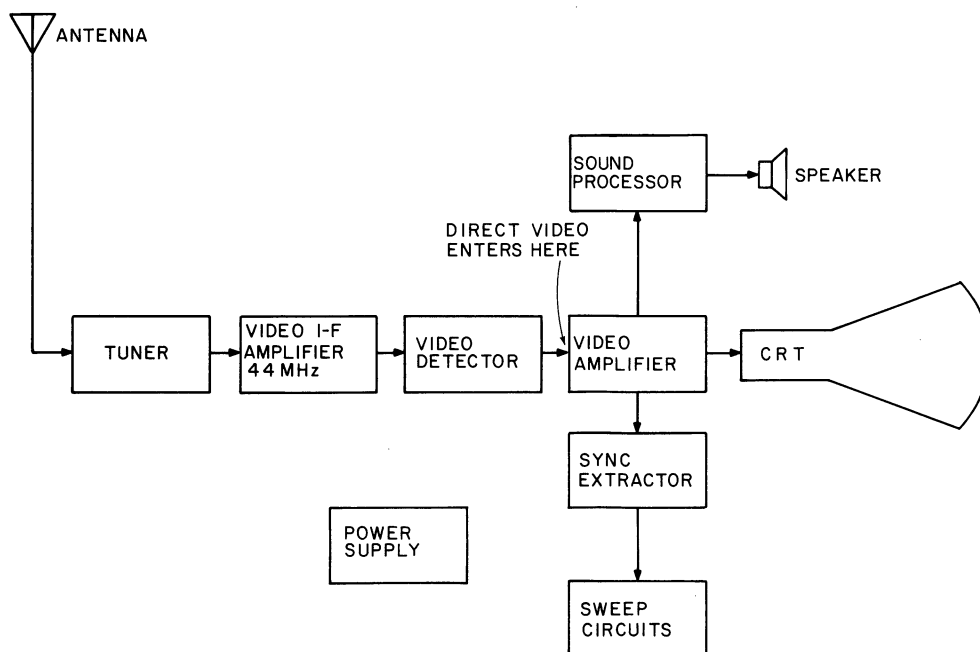
very much improve the display legibility and contrast. Why?

Because we are interested in getting the most legible character of the highest contrast we can. This is not necessarily the one having the sharpest dot rise and fall times. Many things interact to determine the upper video response of a TV display. These include the tuner settings and the i-f response and alignment, the video detector response, video peaking, the sound trap setting, rf cable reflections, and a host of other responses. Many of these stages are underdamped and will ring if fed too sharp a risetime input, giving us a ghosted,

shabby, or washed out character. By reducing the video bandwidth going into the system, we can move the dot matrix energy lower in frequency, resulting in cleaner characters of higher contrast.

For most TV displays, intentional smearing will help the contrast, legibility, and overall appearance. The ultimate limit to this occurs when the dots overlap and become illegible. The

Fig. 3. Block diagram of typical B and W television.



optimum amount of intentional smear is usually the value of capacitance that is needed to just close the inside of a "W" presented to the display.

#### Adding a Video Input

Video inputs are easy to add to the average television set, provided you follow some reasonable cautions. First and foremost, you must have an accurate and complete schematic of the set to be modified, preferably a Sams Photofact or something similar. The first thing to check is the power supply on the set. If it has a power transformer and has the chassis properly safety isolated from the power line, it's a good choice for a TVT monitor. This is particularly

true of recent small screen, solid state portable TV sets. On the other hand, if you have a hot chassis type with one side of the power line connected to the chassis, you should avoid its use if at all possible. If you must use this type of set, be absolutely certain to use one of the safety techniques outlined later in Fig. 8.

A block diagram of a typical TV set appears in Fig. 3. UHF or VHF signals are downconverted in frequency to a video i-f frequency of 44 MHz and then filtered and amplified. The output of the video i-f is transformer coupled to a video detector, most often a small signal germanium diode. The video detector output is filtered to

remove the carrier and then routed to a video amplifier made up of one or more tubes or transistors.

At some point in the video amplification, the black and white signal is split three ways. First, a reduced bandwidth output routes sync pulses to the sync separator stage to lock the set's horizontal and vertical scanning to the video. A second bandpass output sharply filtered to 4.5 MHz extracts the FM sound subcarrier and routes this to a sound i-f amplifier for further processing. The third output is video, which is strongly amplified and then capacitively coupled to the cathode of the picture tube.

The gain of the video amplifier sets the contrast of the display, while the bias setting on the cathode of the picture tube (with respect to its grounded control grid) sets the display brightness. Somewhere in the video amplifier, further rejection of the 4.5 MHz sound subcarrier is usually picked up to minimize picture interference. This is called a sound trap. Sound traps can be a series resonant circuit to ground, a parallel resonant circuit in the video signal path, or simply part of the transformer that is picking off the sound for more processing.

The video detector output is usually around 2 volts peak to peak and usually subtracts from a white level bias setting. The stronger the signal, the more negative the swing, and the blacker the picture. Sync tips are blacker than black, helping to blank the display during retrace times.

Fig. 4 shows us the typical video circuitry of a transistor black and white television. Our basic circuit consists of a diode detector, a unity gain emitter follower, and a variable gain video output stage that is capacitively coupled to the picture tube. The cathode bias sets the brightness, while the video gain sets the contrast. Amplified signals for sync and sound are removed from the collector of the video driver by way of a 4.5 MHz resonant transformer for the sound and a low pass filter for the sync. A parallel resonant trap set to 4.5 MHz eliminates sound interference. Peaking coils on each stage extend the bandwidth by providing higher impedances

and thus higher gain to high frequency video signals.

Note particularly the biasing of the video driver. A bias network provides us with a stable source of 3 volts. In the absence of input video, this 3 volts sets the white level of the display, as well as establishing proper bias for both stages. As an increasing signal appears at the last video output transformer, it is negatively rectified by the video detector, thus lowering the 3 volts proportionately. The stronger the signal, the blacker the picture. Sync will be the strongest of all, giving us a blacker than black bias level of only one volt.

The base of our video driver has the right sensitivity we need for video entry,

accepting a maximum of a 2 volt peak to peak signal. It also has the right polarity, for a positive going bias level means a whiter picture. *But, an unmodified set is already biased to the white level, and if we want to enter our own video, this bias must be shifted to the black level.*

We have a choice in any TV of direct or ac coupling of our input video. Direct coupling is almost always better as it eliminates any

Fig. 4. Typical video circuitry of transistor B and W TV set.

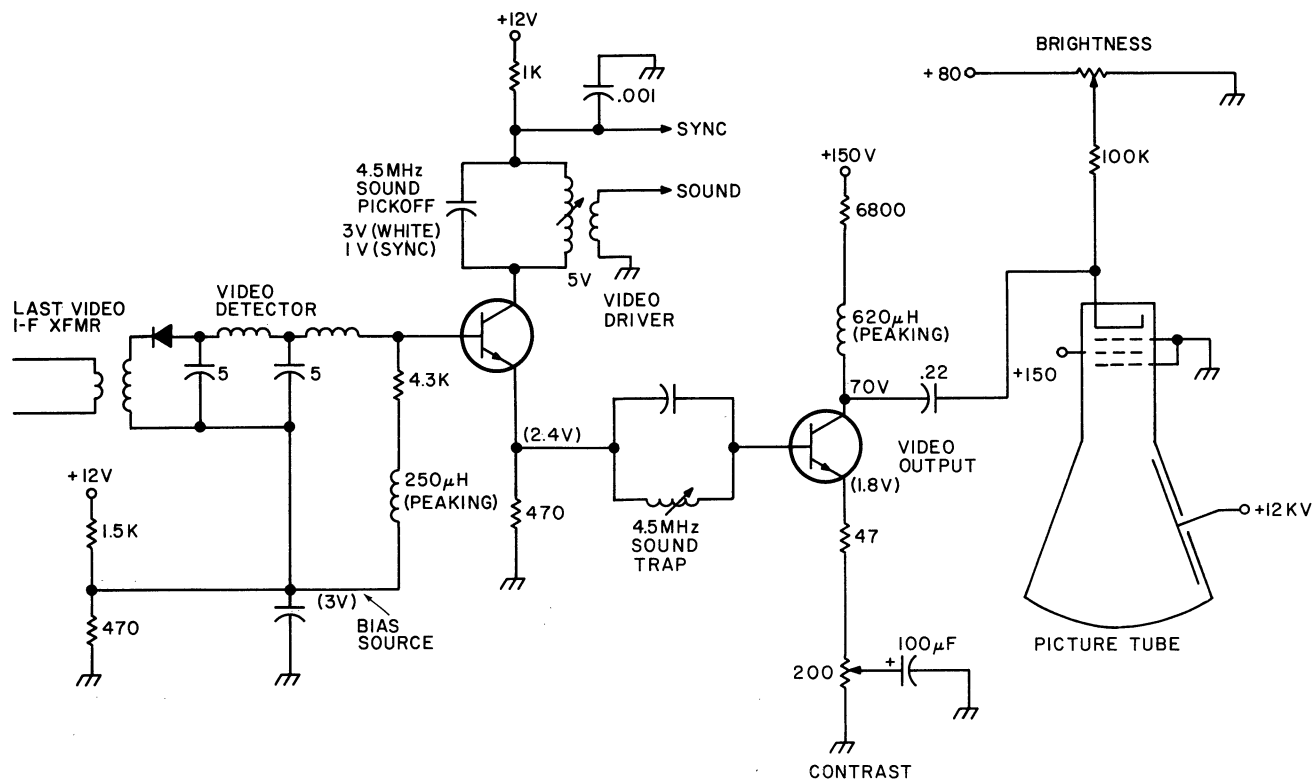


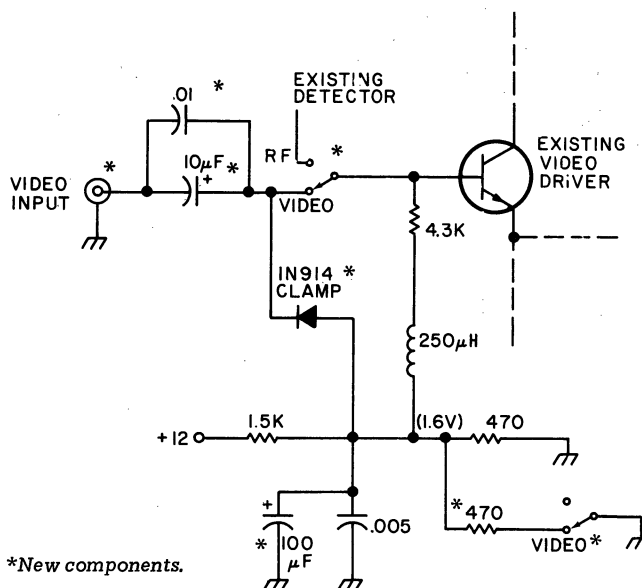
Fig. 5. Direct coupled video uses 1.2 volt offset of Darlington transistor as bias.

shading effects or any change of background level as additional characters are added to the screen. Fig. 5 shows how we can direct couple our video into a transistor black and white set. We provide a video input, usually a BNC or a phono jack, and route this to a PNP Darlington transistor or transistor pair, borrowing around 5 mils from the set's +12 volt supply. This output is routed to the existing video driver stage through a SPDT switch that either picks the video input or the existing video detector and bias network.

The two base-emitter diode drops in our Darlington transistor add up to a 1.2 volt positive going offset; so, in the absence of a video input or at the base of a sync tip, the video driver is biased to a blacker than black sync level of 1.2 volts. With a white video input of 2 volts, the video driver gets biased to its usual 3.2 volts of white level. Thus, our input transistor provides just the amount of offset we need to match the white and black bias levels of our video driver. Note that the old bias network is on the other side of the switch and does nothing in the video position.

Two other ways to offset our video input are to use two ordinary transistors connected in the Darlington configuration, or to use one transistor and a series diode

Fig. 6. Ac coupled video needs shift of bias to black level plus a clamping diode.



to pick up the same amount of offset, as shown in Fig. 5. If more or less offset is needed, diodes or transistors can be stacked up further to pick up the right amount of offset.

The important thing is that the video driver ends up with the same level for white bias and for black bias in either position of the switch.

Ac or capacitively coupled video inputs should be avoided. Fig. 6 shows a typical circuit. The TV's existing bias network is lowered in voltage by adding a new parallel resistor to ground to give us a voltage that is 0.6 volts more positive than the blacker than black sync tip voltage. For instance, with a 3 volt white level, and

2 volt peak to peak video, the sync tip voltage would be 1 volt; the optimum bias is then 1.6 volts. Input video is capacitively coupled by a fairly large electrolytic capacitor in parallel with a good high frequency capacitor. This provides for a minimum of screen shading and still couples high frequency signals properly. A clamping diode constantly clamps the sync tips to their bias value, with the 0.6 volt drop of this diode being taken out by the extra 0.6 volts provided for in the bias network. This clamping diode automatically holds the sync tips to their proper value, regardless of the number of white dots in the picture. Additional bypassing of the bias network by a large electrolytic may be needed for proper operation of the clamping diode, as shown in Fig. 6. Note that our bias network is used in both switch positions — its level is shifted as needed for the direct video input.

Tube type sets present about the same interface problems as the solid state versions do. Fig. 7 shows a typical direct coupled tube interface. In the unmodified

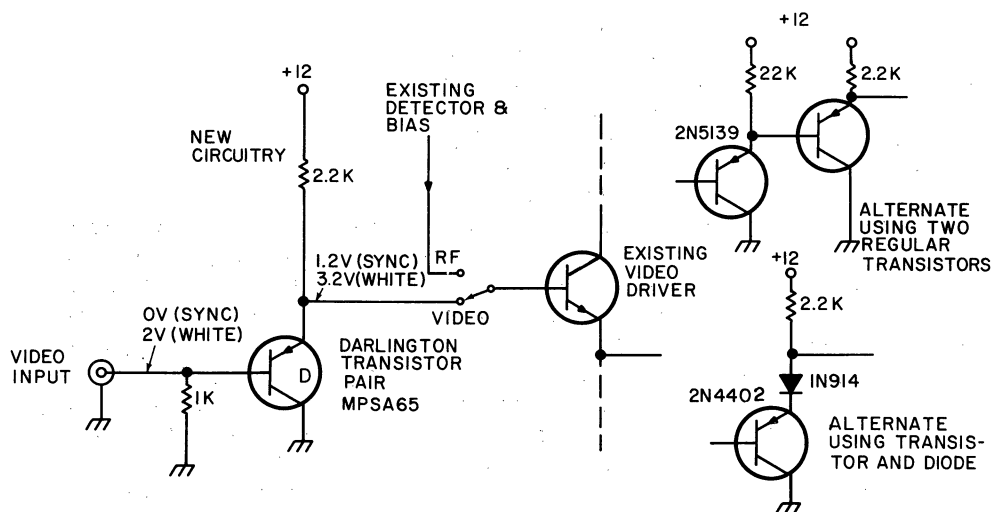
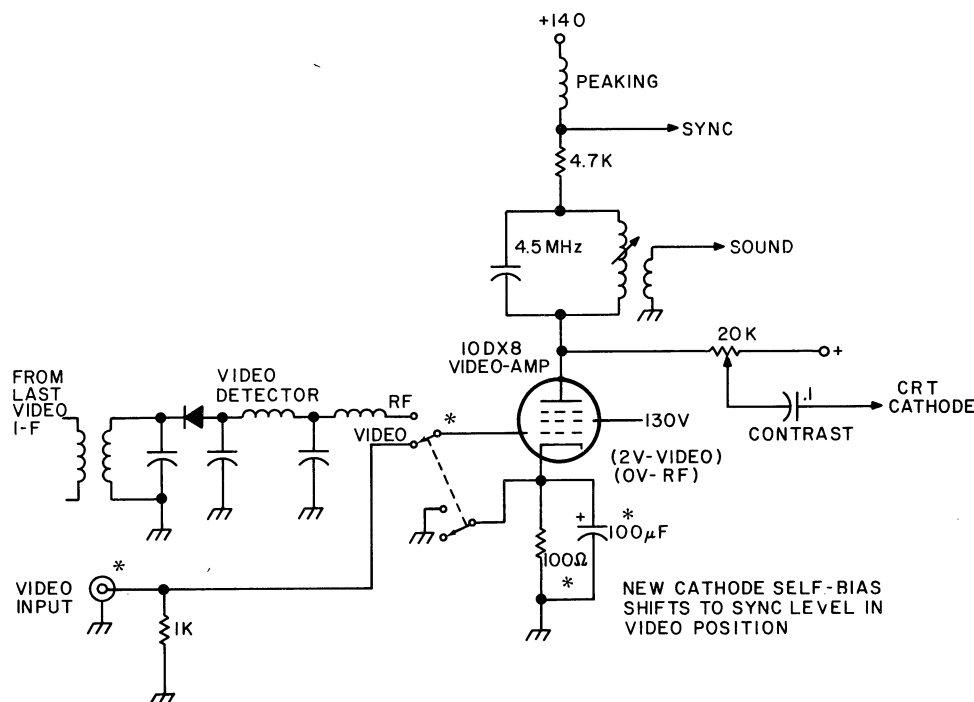


Fig. 7. Direct coupled video added to tube type B and W television.



\*New components.

circuit, the white level is zero volts and the sync tip black level is minus two volts. If we can find a negative supply (scarce in tube type circuits), we could offset our video in the negative direction by two volts to meet these bias levels.

Instead of this, it is usually possible to self bias the video amplifier to a cathode voltage of +2 volts. This is done by breaking the cathode to ground connection and adding a small resistor (50 to 100 Ohms) between cathode and ground to get a cathode voltage of +2 volts. Once this value is found, a heavy electrolytic bypass of 100 microfarads or more is placed in parallel with the resistor. Switching then grounds the cathode in the normal rf mode and makes it +2 volts in the video entry mode.

In the direct video mode, a sync tip grounded input presents zero volts to the grid, which is self biased

minus two volts with respect to the cathode. A white level presents +2 volts to the grid, which equals zero volts grid to cathode.

Should there already be a self bias network on the cathode, it is increased in value as needed to get the black rather than white level bias in the direct video mode.

#### Hot Chassis Problems

There is usually no shock hazard when we use clip-on rf entry or when we use a direct video jack on a transformer-powered TV. A very severe shock hazard can exist if we use direct video entry with a TV set having one side of the

power line connected to the chassis. Depending on which way the line cord is plugged in, there is a 50-50 chance of the hot side of the power line being connected directly to the chassis.

Hot chassis sets, particularly older, power hungry tube versions, should be avoided entirely for direct video entry. If one absolutely must be used, some of the suggestions of Fig. 8 may ease the hazard. These include using an isolation transformer, husky back-to-back filament transformers, three wire power systems, optical coupling of the video input,

and total package isolation. Far and away the best route is simply never to attempt direct video entry onto a hot chassis TV.

### **Making the Conversion**

Fig. 9 sums up how we modify a TV for direct video entry. Always have a complete schematic on hand, and use a transformer style TV set if at all possible. Late models, small screen, medium to high quality solid state sets are often the best display choice. Avoid using junk sets, particularly very old ones. Direct coupling of video is far preferable to ac capacitor coupling. Either method has to maintain the black and white bias levels on the first video amplifier stage. A shift of the first stage quiescent bias from normally white to normally black is also a must. Use short, shielded leads between the video input jack and the rest of the circuit. If a changeover switch is used, keep it as close to the rest of the video circuitry as you possibly can.

### **Extending Video and Display Bandwidth**

By using the direct video input route, we eliminate any bandwidth and response restrictions of an rf

modulator, the tuner, video i-f strip, and the video detector filter. Direct video entry should bring us to a 3 MHz bandwidth for a color set and perhaps 3.5 MHz for a black and white model, unless we are using an extremely bad set. The resultant 6 to 7 million dot per second rate is adequate for short character lines of 32, 40, and possibly 48 characters per line. But the characters will smear and be illegible if we try to use longer line lengths and premium (lots of dots) character generators on an ordinary TV. Is there anything we can do to the set to extend the video bandwidth and display response for these longer line lengths?

In the case of a color TV, the answer is probably no. The video response of a color set is limited by an essential delay line and an essential 3.58 MHz trap. Even if we were willing to totally separate the chrominance and luminance channels, we'd still be faced with an absolute limit set by the number of holes per horizontal line in the shadow mask of the tube. This explains why video color displays are so expensive and so rare. Later on, we'll look at what's involved in adding color to the shorter line lengths.

With a black and white TV, there is often quite a bit

Fig. 8. Getting Around a Hot Chassis Problem.

Hot chassis problems can be avoided entirely by using only transformer-powered TV circuits or by using clip-on rf entry. If a hot chassis set must be used, here are some possible ways around the problem:

#### **1. Add an isolation transformer.**

A 110 volt to 110 volt isolation transformer whose wattage exceeds that of the set may be used. These are usually expensive, but a workable substitute can be made by placing two large surplus filament transformers back to back. For instance, a pair of 24 volt, 4 Amp transformers can handle around 100 Watts of set.

#### **2. Use a three wire system with a solid ground.**

Three prong plug wiring, properly polarized, will force the hot chassis connection to the cold side of the power line. This protection is useful only when three wire plugs are used in properly wired outlets. A severe shock hazard is reintroduced if a user elects to use an adaptor or plugs the system into an unknown or improperly wired outlet. The three wire system should **NOT** be used if anyone but yourself is ever to use the system.

#### **3. Optically couple the input video.**

Light emitting diode-photocell pairs are low in cost and can be used to optically couple direct video, completely isolating the video input from the hot chassis. Most of these optoelectronic couplers do not have enough bandwidth for direct video use; the Litronix IL-100 is one exception. Probably the simplest route is to use two separate opto-isolators, one for video and one for sync, and then recombine the signals inside the TV on the hot side of the circuit.

#### **4. Use a totally packaged and sealed system.**

If you are only interested in displaying messages and have no other input/output devices, you can run the entire circuit hot chassis, provided everything is sealed inside one case and has no chassis-to-people access. Interface to teletypes, cassettes, etc., cannot be done without additional isolation, and servicing the circuit presents the same shock hazards that servicing a hot chassis TV does.



we can do to present long lines of characters, depending on what set you start out with and how much you are willing to modify the set.

The best test signal you can use for bandwidth extension is the dot matrix data you actually want to display, for the frequency response, time delay, ringing, and overshoot all get into the act. What we want to end up with is a combination that gives us reasonably legible characters.

A good oscilloscope (15 MHz or better bandwidth) is very useful during bandwidth extension to show where the signal loses its response in the circuit. At any time during the modification process, there is usually one response bottleneck. This, of course, is what should be attacked first. Obviously the better a TV you start with, the easier will be the task. Tube type gutless wonders, particularly older ones, will be much more difficult to work with than with a modern, small screen, quality solid state portable.

Several of the things we can do are watching the control settings, getting rid of the sound trap, minimizing circuit strays, optimizing spot size, controlling peaking, and shifting to higher current operation. Let's take a look at these in turn.

#### Control Settings

Always run a data display at the lowest possible contrast and using only as much brightness as you really need. In many circuits, low contrast means a lower video amplifier gain, and thus less of a gain-bandwidth restriction.

#### Eliminate the Sound Trap

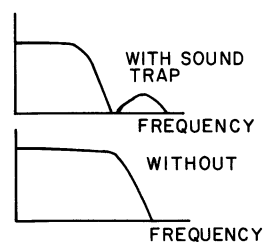
The sound trap adds a notch at 4.5 MHz to the video response. If it is eliminated or switched out of the circuit, a wider video bandwidth automatically

Fig. 9. How to Add a Direct Video Input to a TV Set.

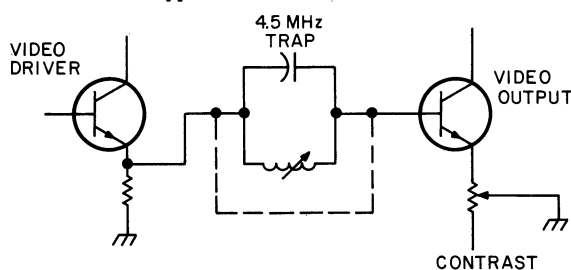
1. Get an accurate and complete schematic of the set — either from the manufacturer's service data or a Photofact set. **Do not try adding an input without this schematic!**
2. Check the power supply to see if a power transformer is used. If it is, there will be no shock hazard, and the set is probably a good choice for direct video use. If the set has one side of the power line connected to the chassis, a severe shock hazard exists, and one of the techniques of Fig. 8 should be used. **Avoid the use of hot chassis sets.**
3. Find the input to the first video amplifier stage. Find out what the white level and sync level bias voltages are. The marked or quiescent voltage is usually the white level; sync is usually 2 volts less. A transistor TV will typically have a +3 volt white level and a +1 volt sync level. A tube type TV will typically have a zero volt white level and a -2 volt sync level.
4. Add a changeover switch using minimum possible lead lengths. Add an input connector, either a phono jack or the premium BNC type connector. Use shielded lead for interconnections exceeding three inches in length.
5. Select a circuit that couples the video and biases the first video amplifier stage so that the white and sync levels are preserved. For transistor sets, the direct coupled circuits of Fig. 5 may be used. For tube sets, the circuit of Fig. 7 is recommended. Avoid the use of ac coupled video inputs as they may introduce shading problems and changes of background as the screen is filled.
6. Check the operation. If problems with contrast or sync tearing crop up, recheck and adjust the white and sync input levels to match what the set uses during normal rf operation. Note that the first video stage must be biased to the white level during rf operation and to the sync level for direct video use. The white level is normally two volts more positive than the sync level.

Fig. 10. Removing the sound trap can extend video bandwidth.

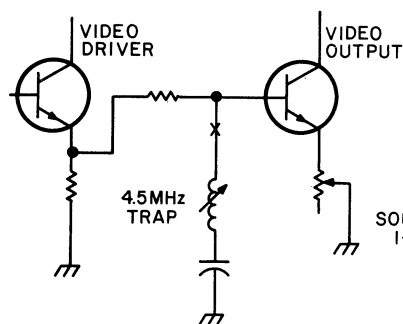
(a) Response



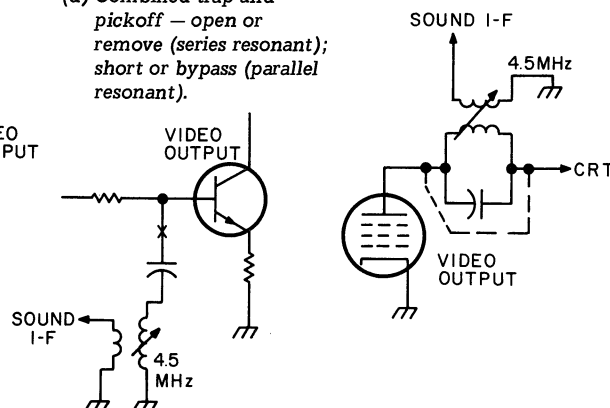
(b) Parallel resonant trap — short or bypass.



(c) Series resonant trap — open or remove.



(d) Combined trap and pickoff — open or remove (series resonant); short or bypass (parallel resonant).



results. Fig. 10 shows us the response changes and the several positions for this trap. Generally, series resonant traps are opened and parallel resonant traps are shorted or bypassed through suitable switching or outright elimination. The trap has to go back into the circuit if the set is ever again used for ordinary program reception. Sometimes simply backing the slug on the trap all the way out will improve things enough to be useful.

### Minimizing Strays

One of the limits of the video bandwidth is the stray capacitance both inside the video output stage and in the external circuitry. If the contrast control is directly in the signal path and if it has long leads going to it, it may be hurting the response. If you are using the TV set exclusively for data display, can you rearrange the control location and simplify and shorten the video output to picture tube interconnections?

### Additional Peaking

Most TV sets have two peaking networks. The first of these is at the video detector output and compensates for the vestigial sideband transmission signal that makes sync and other

low frequency signals double the amplitude of the higher frequency ones. The second of these goes to the collector or plate of the video output stage and raises the circuit impedance and thus the effective gain for very high

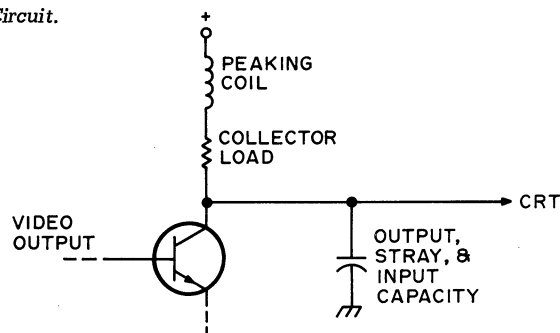
frequencies. Sometimes you can alter this second network to favor dot presentations. Fig. 11 shows a typical peaking network and the effects of too little or too much peaking. Note that the stray capacitance also enters into the peaking, along with the video amplifier output capacitance and the picture tube's input capacitance. Generally, too little peaking will give you low contrast dots, while too much will give you sharp dots together and shift the more continuous portions of the characters objectionably. Peaking is changed by increasing or decreasing the series inductor from its design value.

### Running Hot

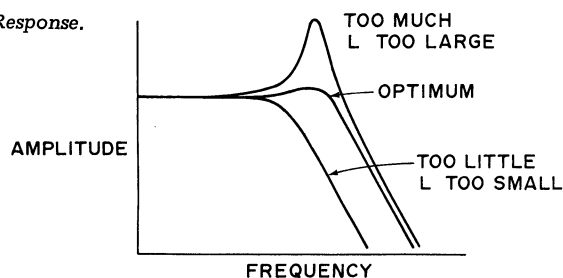
Sometimes increasing the operating current of the video output stage can increase the system bandwidth — IF this stage is in fact the limiting response, IF the power supply can handle the extra current, IF the stage isn't already parked at its gain-bandwidth peak, and IF the extra heat can be gotten rid of without burning anything up. Usually, you can try adding a resistor three times the plate or collector load resistor in parallel, and see if it increases bandwidth by 1/3. Generally, the higher the current, the wider the bandwidth, but watch

Fig. 11. Adjusting the peaking coil can extend video response.

(a) Circuit.



(b) Response.



carefully any dissipation limits. Be sure to provide extra ventilation and additional heatsinking, and check the power supply for unhappiness as well. For major changes in operating current, the emitter resistors and other biasing components should also be proportionately reduced in value.

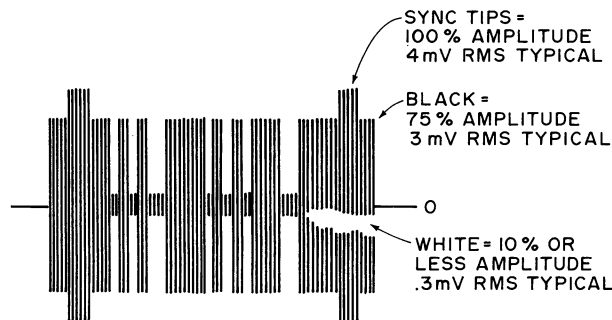
### Spot Size

Even with excellent video bandwidth, if you have an out-of-focus, blooming, or changing spot size, it can completely mask character sharpness. Spot size ends up the ultimate limit to resolution, regardless of video bandwidth.

Once again, brightness and contrast settings will have a profound effect, with too much of either blooming the spot. Most sets have a focus jumper in which ground or a positive voltage is selected. You can try intermediate values of voltage for maximum sharpness. Extra power supply filtering can sometimes minimize hum and noise modulation of the spot.

Anything that externally raises display contrast will let you run with a smaller beam current and a sharper spot. Using circularly polarized filters, graticule masks, or simple colored filters can

Fig. 13. Standard rf interface levels. Impedance = 300Ω. Carrier frequency per Fig. 14.



minimize display washout from ambient lighting. Fig. 12 lists several sources of material for contrast improvement. Much of this is rather expensive, with pricing from \$10 to \$25 per square foot being typical. Simply adding a hood and positioning the display away from room lighting will also help and is obviously much cheaper.

### Direct Rf Entry

If we want the convenience of a "free" display, the freedom from hot chassis problems, and "use it anywhere" ability, direct rf entry is the obvious choice. Its two big limitations are the need for FCC type approval, and a limited video bandwidth that in turn limits the number of characters per line and the number of dots per character.

An rf interface standard is shown in Fig. 13. It consists of an amplitude modulated carrier of one of the standard television channel video frequencies of Fig. 14. Channel 2 is most often used

with a 55.250 MHz carrier frequency, except in areas where a local commercial Channel 2 broadcast is intolerably strong. Circuit cost, filtering problems, and stability problems tend to increase with increasing channel number.

The sync tips are the strongest part of the signal, representing 100% modulation, often something around 4 millivolts rms across a 300 Ohm line. The black level is 75% of the sync level, or about 3 millivolts for 4 millivolt sync tips. White level is less than 10% of maximum. Note that the signal is weakest when white and strongest when sync. This is the exact opposite of the video interface of Fig. 1.

Rf modulators suitable for clip-on rf entry TV typewriter use are called Class 1 TV Devices by the FCC. A Class 1 TV device is supposed to meet the rules and regulations summarized in Fig. 15.

Fig. 16 shows us a block diagram of the essential parts of a TV modulator. We start

Fig. 14. Television Picture Carrier Frequencies.

Channel 2	55.25 MHz
Channel 3	61.25 MHz
Channel 4	67.25 MHz
Channel 5	77.25 MHz
Channel 6	83.25 MHz

Fig. 15. FCC Regulations on Class 1 TV Devices. More complete information appears in subpart H of Part 15 and subpart F of Part 2 of the Federal Communications Commission Rules and Regulations. It is available at many large technical libraries.

A Class 1 TV device generates a video modulated rf carrier of a standard television channel frequency. It is directly connected to the antenna terminals of the TV set.

The maximum rms rf voltage must be less than 6 millivolts using a 300 Ohm output line.

The maximum rf voltage on any frequency more than 3 MHz away from the operating channel must be more than 30 dB below the peak in-channel output voltage.

An antenna disconnect switch of at least 60 dB attenuation must be provided.

No user adjustments are permitted that would exceed any of the above specifications.

Residual rf radiation from case, leads and cabinet must be less than 15 microvolts per meter.

A Class 1 TV device must not interfere with TV reception.

Type approval of the circuit is required. A filing fee of \$50 and an acceptance fee of \$250 is involved.

Fig. 12. Contrast Enhancing Filter Materials.

### Circularly polarized filters:

**Polaroid Corp.**  
Cambridge MA 02139

### Anti-reflection filters:

**Panelgraphic Corp.**  
10 Henderson Dr.  
West Caldwell NJ 07006

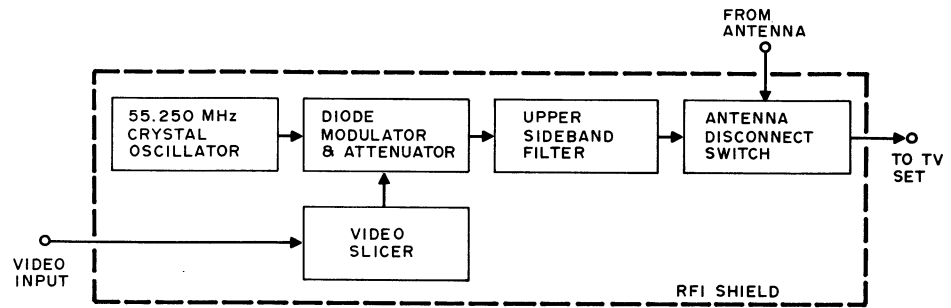
### Light control film:

**3M Visual Products Div.**  
3M Center  
St. Paul MN 55101

### Acrylic plexiglas filter sheets:

**Rohm and Haas**  
Philadelphia PA 19105

Fig. 16. Block diagram of rf modulator.



with a stable oscillator tuned to one of the Fig. 14 frequencies. A crystal oscillator is a good choice, and low cost modules are widely available. The output of this oscillator is then amplitude modulated. This can be done by changing the bias current through a silicon small signal diode. One milliampere of bias current makes the diode show an ac and rf impedance of 26 Ohms. Half a mil will look like 52 Ohms, and so on. The diode acts as a variable resistance attenuator in the rf circuit, whose bias is set and changed by the video circuit.

Since diode modulators are non-linear, we can't simply apply a standard video signal to them and get a standard rf signal out. A differential amplifier circuit called a video slicer may be used to compensate for this non-linearity. The video slicer provides three distinct currents to the diode modulator. One of these is almost zero for the white level, while the other two provide the black and sync levels. A contrast control that sets the slicing level lets you adjust the sync tip height with respect to the black level. The video slicer also minimizes rf getting back into the video. An attenuator to reduce the size of the modulated signal usually follows the diode modulator.

An upper side band filter removes most of the lower sideband from the AM modulated output, giving us a

vestigial sideband signal that stays inside the channel band limits. This same filter eliminates second harmonic effects and other spurious noise. The filter's output is usually routed to an antenna disconnect switch and the TV's antenna terminals. A special switch is needed to provide enough isolation.

Some of the actual circuitry involved is shown in Fig. 17. The video slicer consists of a pair of high gain, small signal NPN transistors, while the oscillator is a commercially available module.

Rf entry systems always must be direct coupled to the antenna terminals of the set and should never provide any more rf than is needed for a minimum snow-free picture. They should be permanently tuned to a single TV channel. Under no circumstances should an antenna or cable service hookup remain connected to the set during TVT use, nor should radiation rather than a direct rf cable connection ever be used.

#### Color Techniques

We can add a full color capability to a TV typewriter system fairly easily and cheaply — provided its usual

black and white video dot rate is low enough in frequency to be attractively displayed on an ordinary color TV. Color may be used to emphasize portions of a message, to attract attention, or as obvious added value to a graphics display. Color techniques work best on TV typewriter systems having a horizontal frequency very near 15,735 Hertz.

All we basically have to do is generate a subcarrier sine wave to add to the video

output. The phase of this subcarrier (or its time delay) is shifted with respect to what the phase was immediately after each horizontal sync pulse to generate the various colors.

Fig. 18 shows us the differences between normal color and black and white operation. Black and white baseband video is some 4 MHz wide and has a narrow 4.5 MHz sound subcarrier. The video is amplitude modulated, while the sound is narrow band frequency

Fig. 17. Channel two oscillator, modulator, video slicer and attenuator. R sets output level.

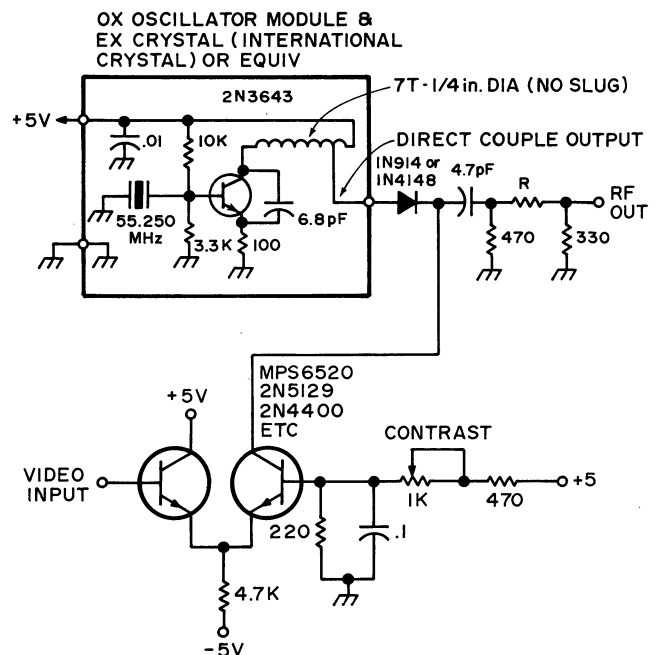
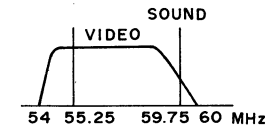
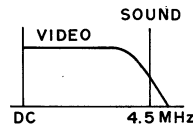


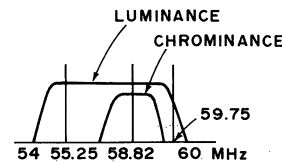
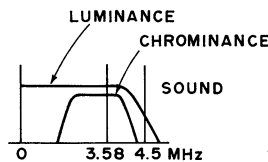
Fig. 18. Differences between color and black and white spectra.

(a) Black and white — baseband video.



(b) Black and white — Channel two rf.

(c) Color — baseband video.



(d) Color — Channel two rf.

modulated. This translates up to a 6 MHz rf channel with a vestigial lower sideband as shown in Fig. 18(b).

To generate color, we add a new pilot or subcarrier at a magic frequency of 3.579545 MHz — see Fig. 18(c). What was the video is now called the luminance, and is the same as the brightness in a black and white system. The new subcarrier and its modulation is called the chrominance signal and determines what color gets displayed and how saturated the color is to be.

Since the black and white information is a sampled data system that is scanned at the vertical and horizontal rates, there are lots of discrete holes in the video spectrum that aren't used. The color subcarrier is designed to stuff itself into these holes (exactly in a NSTC color system, and pretty much in a TVT display). Both chrominance and luminance signals use the

same spectral space, with the one being where the other one isn't, overlapping comb style.

The phase or relative delay of the chrominance signal with respect to a reference determines the instantaneous color, while the amplitude of this signal with respect to the luminance sets the saturation of the color. Low amplitudes generate white or pastel shades, while high amplitudes of the chrominance signal produce saturated and deep colors.

At least eight cycles of a reference or burst color phase are transmitted immediately following each horizontal sync pulse as a timing reference, as shown in Fig.

19. The burst is around 25% of maximum amplitude, or about the peak to peak height of a sync pulse.

The TV set has been trained at the factory to sort all this out. After video detection, the set splits out the chrominance channel with a bandpass amplifier and then synchronously demodulates it with respect to an internal 3.58 MHz reference. The phase of this demodulation sets the color and the amplitude sets the saturation by setting the

ratios of electron beam currents on the picture tube's red, blue and green guns.

Meanwhile, the luminance channel gets amplified as brightness style video. It is delayed with a delay line to make up for the time delay involved in the narrower band color processing channel. It is then filtered with two traps — the 4.5 MHz sound trap, and a new trap to get rid of any remaining 3.58 MHz color subcarrier that's left. The luminance output sets the overall brightness by modulating the cathodes of all three color guns simultaneously.

Just after each horizontal sync pulse, the set looks for the reference burst and uses this reference in a phase

Fig. 19 Adding a color reference burst to the back porch of the horizontal sync pulses.

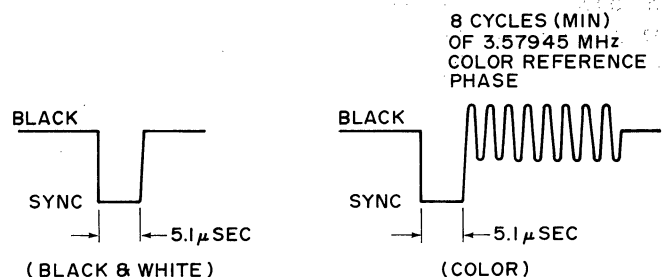


Fig. 20. Colors Are Generated by Delaying or Phase Shifting the Burst Frequency.

Color	Approximate Phase	Approximate Delay
Burst	0°	0
Yellow	15°	12 nanoseconds
Red	75°	58 nanoseconds
Magenta	135°	105 nanoseconds
Blue	195°	151 nanoseconds
Cyan	255°	198 nanoseconds
Green	315°	244 nanoseconds

wanted. The output chrominance signal is RC filtered to make it somewhat sinusoidal. It's then cut down in amplitude to around one-quarter the maximum video white level and is capacitively coupled to the 100 Ohm video output of Fig. 2 or otherwise summed into the video or rf modulator circuitry. For truly dramatic color effects, the amplitude and delay of the chrominance signal can be changed in a more complex version of the same circuit.

More information useful in solving television interface appears in the *Television Engineering Handbook*, by Donald Fink, and in various issues of the *IEEE Transactions on Consumer Electronics*.

detector circuit to keep its own 3.58 MHz reference locked to the version being transmitted.

Fig. 20 shows us the phase angles related to each color with respect to the burst phase. It also shows us the equivalent amount of delay we need for a given phase angle. Since we usually want only a few discrete colors, it's far easier to digitally generate colors simply by delaying the reference through gates or buffers, rather than using complex and expensive analog phase shift methods.

Strictly speaking, we should control both the chrominance phase and amplitude to be able to do both pastel and strongly saturated colors. But simply keeping the subcarrier amplitude at the value we used for the burst — around 25% of video amplitude — is far simpler and will usually get us useful results.

A circuit to add color to a TV typewriter is shown in Fig. 21. A 3.579545 MHz crystal oscillator drives a string of CMOS buffers that make up a digital delay line. The output delays caused by the propagation delay times in each buffer can be used as

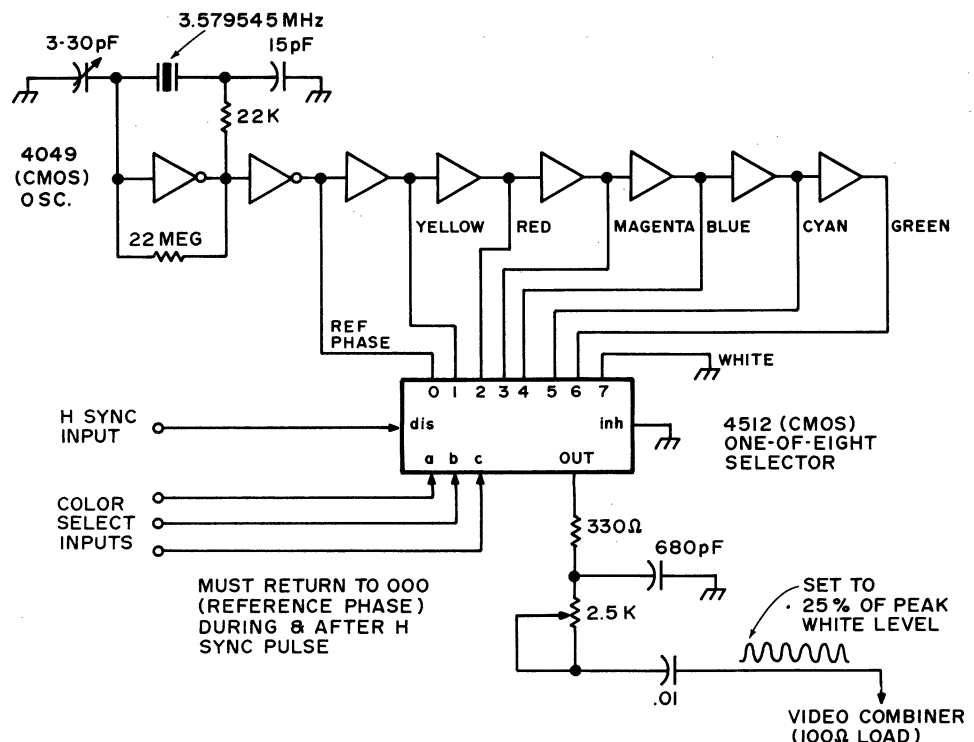
is, or can be trimmed to specific colors by varying the supply voltage.

The reference phase and the delayed color outputs go to a one-of-eight data selector. The data selector picks either the reference or a selected color in response to a code presented digitally to the three select lines. The logic that is driving this selector must return to the

reference phase position (000) immediately before, during and for a minimum of a few microseconds after each horizontal sync pulse. This gives the set a chance to lock and hold onto the reference color burst.

The chrominance output from the data selector should be disabled for the duration of the sync pulses and any time a white screen display is

Fig. 21. Color subcarrier generator. Hex buffer used as delay line. Use supply voltage variation on 4050 to trim colors.





Pinouts: Parallel Data Interface (PDI) as  
used on Processor Tech. Sol System

Sept. 30, 1976

MASTER UNIT-Male connector

J2 Pin #	Signal mnemonic	Signal name	J2 pin#	Signal mnemonic	Signal name
1	CG	Chassis Ground	14	US	Unit Select
2	SG	Signal Ground	15	OE	Output Enable
3	IE	Input Enable	16	$\overline{\text{XDR}}$	$\overline{\text{eXternal Device Ready}}$
4	$\overline{\text{DR}}$	$\overline{\text{Data Ready}}$	17	$\overline{\text{OL}}$	$\overline{\text{Output Load}}$
5	I $\overline{\text{AK}}$	$\overline{\text{Input Acknowledge}}$	18	OD7	Output Data, bit 7
6	ID7	Input Data, bit 7	19	OD6	Output Data, bit 6
7	ID6	Input Data, bit 6	20	OD5	Output Data, bit 5
8	ID5	Input Data, bit 5	21	OD4	Output Data, bit 4
9	ID4	Input Data, bit 4	22	OD3	Output Data, bit 3
10	ID3	Input Data, bit 3	23	OD2	Output Data, bit 2
11	ID2	Input Data, bit 2	24	OD1	Output Data, bit 1
12	ID1	Input Data, bit 1	25	OD0	Output Data, bit 0
13	ID0	Input Data, bit 0			

Pinouts: Serial Data Interface (SCI) as  
used on Processor Tech. Sol System

Female connector-DB25S

J1 pin#	Signal mnemonic	Signal name	J1 pin#	Signal mnemonic	Signal name
1	CG	Chassis Ground	8	CD	Carrier Detect
2	TD	Transmit Data	11	CLO	Current Loop Output
3	RD	Receive Data	12	LR1	Loop Receiver 1
4	RTS	Request To Send	13	LR2	Loop Receiver 2
5	CTS	Clear To Send	20	DTR	Data Terminal Ready
6	DSR	Data Set Ready	23	LCS	Loop Current Source
7	SG	Signal Ground			

Note 1: Many pins not specified here are used in EIA RS-232C specification.  
USE THEM WITH CAUTION.

Note 2: Terminals output on pins 2, 4 & 20 and input on pins 3, 5 & 6 for EIA type hookups. Modems and computer mainframes output on pins 3, 5 & 6 and input on pins 2, 4 & 20.

Note 3: Current loop hookups are the same for terminals, modems, mainframes.

## J3 Keyboard Connector (between U64 and U65)

Sol-PC, Rev. 2,E  
10/18/76

pin no.	Signal name	pin no.	Signal name
1	ground	11	ground
2	+5v	12	+5v
3	Kbd Data Ready	13	Restart
4	Break	14	Local
5	Kbd Data 0	15	KBd Data 4
6	Kbd Data 1	16	KBd Data 5
7	Kbd Data 2	17	KBD Data 6
8	Kbd Data 3	18	KBD Data 7
9	+5v	19	+5v
10	ground	20	ground

## J4 Display Expansion Connector (between U28, 29)

pin no.	Signal name	pin no.	Signal name
1	ground	11	ground
2	N.C.	12	N.C.
3	Char. addr. 4	13	Dot Clock, 14.318MHz
4	Character clock	14	Composite sync. out
5	Char. addr. 0	15	TTL Serial Data Out
6	Char. addr. 1	16	Composite blanking out
7	Char. addr. 2	17	Scan advance out
8	Char. addr. 3	18	Char. addr. 5
9	N.C.	19	N.C.
10	ground	20	ground

## J5 Personality Module Edge Connector

pin no.	Signal name	pin no.	Signal name
B15	B Ground	A15	Ground
B14	O +5VDC	A14	+5VDC
B13	T Addr. 9	A13	T Addr. 0
B12	T Addr. 8	A12	O Addr. 4
B11	T Addr. 7	A11	P Addr. 3
B10	O INT Bus 0	A10	P Addr. 2
B9	M INT Bus 1	A9	R Addr. 1
B8	M INT Bus 2	A8	O Addr. 5
B7	R INT Bus 3	A7	W Addr. 6
B6	O INT Bus 4	A6	W C4
B5	W INT Bus 5	A5	P C0
B4	W Program 0	A4	P INT Bus 6
B3	P Program 1	A3	I INT Bus 7
B2	P Program 2	A2	N -12VDC
B1	I Program 3	A1	S +12VDC
	N		
	S		

J6 Audio Out for CUTS Cassette Interface: Mini-phone jack at rear panel

J7 Audio In for CUTS Cassette Interface: Mini-phone jack at rear panel

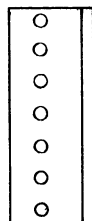
J8 Tape Motor Control 1: (See output port FA, bit 7) Sub-mini jack at rear panel

J9 Tape Motor Control 2: (See output port FA, bit 6) Sub-mini jack at rear panel

Rev A

# J10 DC Power Connector, Sol-PC

Ground  
+5VDC  
-12 VDC  
+12 VDC  
-12 VDC  
+5 VDC  
Ground



## S-100 Bus Definitions

PIN NUMBER	SYMBOL	NAME	FUNCTION
1	+8V	+8 Volts	Unregulated voltage on bus, supplied to PC boards and regulated to 5V supplied by Sol-20 supply
2	-16V	-16 Volts	Positive unregulated voltage supplied by Sol-20 power supply
3	XRDY	EXTERNAL READY	External ready input to CPU ready circuitry
4	VI0	Vectored Interrupt Line #0	
5	VI1	Vectored Interrupt Line #1	
6	VI2	Vectored Interrupt Line #2	
7	VI3	Vectored Interrupt Line #3	
8	VI4	Vectored Interrupt Line #4	
9	VI5	Vectored Interrupt Line #5	
10	VI6	Vectored Interrupt Line #6	
11	VI7	Vectored Interrupt Line #7	
12	XRDY2	EXTERNAL READY #2	not used by Sol-PC
13			
to			
17			
18	STAT DSB	STATUS DISABLE	-Allows the buffers for the 8 status lines to be tri-stated
19	C/C DSB	COMMAND/CONTROL DISABLE	-Allows the buffers for the 6 output command/control lines to be tri-stated
20	UNPROT	UNPROTECT	-not used by Sol-PC electronics
21	SS	SINGLE STEP	-not used by Sol-PC
22	ADD DSB	ADDRESS DISABLE	-Allows the buffers for the 16 address lines to be tri-stated
23	DO DSB	DATA OUT DISABLE	-Allows the buffers for the 8 data output lines to be tri-stated
24	Ø2	PHASE 2 CLOCK	
25	Ø1	PHASE 1 CLOCK	
26	PHLDA	HOLD ACKNOWLEDGE	Processor command/control output signal that appears in response to the HOLD signal; indicates that the data and address bus will go to the high impedance state and processor will enter HOLD state after completion of the current machine cycle.

## S-100 Bus Definitions-continued

<u>PIN NUMBER</u>	<u>SYMBOL</u>	<u>NAME</u>	<u>FUNCTION</u>
27	PWAIT	WAIT	-Processor command/control signal that appears in response to the HOLD signal; indicates that the data and address bus will go to the high impedance state and processor will enter HOLD state after completion of the current machine cycle
28	PINTE	INTERRUPT ENABLE	-Processor command/control output signal; indicates interrupts are enabled, as determined by the contents of the CPU internal interrupt flip-flop. When the flip-flop is set (Enable Interrupt instruction), interrupts are accepted by the CPU; when it is reset (Disable Interrupt instruction), interrupts are inhibited.
29	A5	Address Line #5	
30	A4	Address Line #4	
31	A3	Address Line #3	
32	A15	Address Line #15 (MSB)	
33	A12	Address Line #12	
34	A9	Address Line #9	
35	DIO1	Data In/Out line #1	same as pin 94
36	DIO0	Data In/Out line #0	same as pin 95
37	A10	Address Line #10	
38	DIO4	Data In/Out Line #4	same as pin 91
39	DIO5	Data In/Out Line #5	same as pin 92
40	DIO6	Data In/Out Line #6	same as pin 93
41	DIO2	Data In/Out Line #2	same as pin 88
42	DIO3	Data In/Out Line #3	same as pin 89
43	DIO7	Data In/Out Line #7	same as pin 90
44	SM1	MACHINE CYCLE 1	-Status output signal that indicates that the processor is in the fetch cycle for the first byte of an instruction
45	SOUT	OUTPUT	-Status output signal that indicates the address bus contains the address of an output device and the data bus will contain the output data when PWR is active
46	SINP	INPUT	-Status output signal that indicates the address bus contains the address of an input device and the input data should be placed on the data bus when PDBIN is active
47	SMEMR	MEMORY READ	-Status output signal that indicates the data bus will be used to read memory data
48	SHLTA	HALT ACKNOWLEDGE	-Status output signal that acknowledges a HALT instruction
49	<u>CLOCK</u>	<u>CLOCK</u>	- Inverted output of the 02 CLOCK
50	GND	GROUND	
51	+8V	+8 Volts	Unregulated input to 5 volt regulators supplied by Sol-20 power supply
52	-16V	-16 Volts	Negative unregulated voltage supplied by Sol-20 power supply

## S-100 Bus Definitions-continued

PIN NUMBER	SYMBOL	NAME	FUNCTION
53	<u>SSWI</u>	SENSE SWITCH INPUT	not used by Sol
54	<u>EXT CLR</u>	EXTERNAL CLEAR	not used by Sol-PC electronics
55	<u>RTC</u>	REAL TIME CLOCK	not used by Sol-PC electronics
56	<u>STSTB</u>	STATUS STROBE	not used by Sol
57	<u>DIGI</u>	DATA INPUT GATE #1	When low forces PDBINS low and forces CPU input multiplexers to the DIO bus. During CPU DBIN cycle, disables CPU DIO bus drivers
58	FRDY	FRONT PANEL READY	-When low disables MWRITE driver
59	TO BE DEFINED		
64			
65	<u>MREQ</u>	MEMORY REQUEST	- Z 80 signal not used by Sol-PC electronics
66	<u>REF</u>	REFRESH	- Z 80 signal not used by Sol-PC electronics
67	<u>PHANTOM</u>	PHANTOM DISABLE	-Output from CPU section used to disable RAM or ROM during power on initialization program execution
68	MWRITE	MEMORY WRITE	-Indicates that the data present on the Data Out Bus is to be written into the memory location currently on the address bus
69	<u>PS</u>	PROJECT STATUS	-not used by Sol-PC electronics
70	<u>PROT</u>	PROTECT	-not used by Sol-PC electronics
71	<u>RUN</u>	RUN	-not used by Sol-PC electronics
72	<u>PRDY</u>	PROCESSOR READY	- Memory and I/O input to the CPU Board wait circuitry
73	<u>PINT</u>	INTERRUPT REQUEST	- The processor recognizes an interrupt request on this line at the end of the current instruction or while halted. If the processor is in the HOLD state or the Interrupt Enable flip-flop is reset, it will not honor the request.
74	<u>PHOLD</u>	HOLD	-Processor command/control input signal that requests the processor enter the HOLD state; allows an external device to gain control of address and data buses as soon as the processor has completed its use of these buses for the current machine cycle
75	<u>PRESET</u>	RESET	-Processor command/control input; while activated, the content of the program counter is cleared and the instruction register is set to 0
76	PSYNC	SYNC	-Processor command/control output; provides a signal to indicate the beginning of each machine cycle
77	<u>PWR</u>	WRITE	-Processor command/control output; used for memory write or I/O output control. Data on the data bus is stable while the PWR is active
78	PDBIN	DATA BUS IN	-Processor command/control output; indicates to external circuits that the data bus is in the input mode

## S-100 Bus Definitions-continued

PIN NUMBER	SYMBOL	NAME	FUNCTION (LSB)
79	A0	Address Line #0	
80	A1	Address Line #1	
81	A2	Address Line #2	
82	A6	Address Line #6	
83	A7	Address Line #7	
84	A8	Address Line #8	
85	A13	Address Line #13	
86	A14	Address Line #14	
87	A11	Address Line #11	
88	DI02	Data In/Out Line #2	same as pin 41
89	DI03	Data In/Out Line #3	same as pin 42
90	DI07	Data In/Out Line #7	same as pin 43
91	DI04	Data In/Out Line #4	same as pin 38
92	DI05	Data In/Out Line #5	same as pin 39
93	DI06	Data In/Out Line #6	same as pin 40
94	DI01	Data In/Out Line #1	same as pin 35
95	DIO0	Data In/Out Line #0	same as pin 36
96	SINTA	INTERRUPT ACKNOWLEDGE	-Status output signal; acknowledges signal for INTERRUPT request
97	SWO	WRITE OUT	-Status output signal; indicates that the operation in the current machine cycle will be a WRITE memory or output function
98	SSTACK	STACK	-Status output signal indicates that the address bus holds the pushdown stack address from the Stack Pointer
99	POC	POWER-ON CLEAR	
100	GND	GROUND	

---

 SWITCH FUNCTION DEFINITION -- Display Ctrl---Schematic Drawing #4

Switch No.	Mnemonic	ON	OFF
S1-1	RST	Restart to Zero	RUN (Dwg. #1)
S1-2	not used		
S1-3	BLANK	Blank Ctrl Characters	Display Ctrl Char.
S1-4	Polarity		
S1-5	BLINK	Blinking cursor	*Solid or NO cursor
S1-6	SOLID	Solid cursor	*Blinking or NO cursor

\*NO cursor if S1-5 and S1-6 are off at same time.

Both switches should not be on at the same time.

---

 Drawing #3 -- Sense Switch

Switch No.	Mnemonic	ON	OFF
S2-1	SSW0	LSB, data bit	0=LO HI
S2-2thruS2-7		etc.	LO HI
S2-8	SSW7	MSB data bit 7	LO HI



# SERIAL I/O BAUD RATE SWITCH -- Schematic Drawing #3

Switch No.	Mnemonic	ON	OFF
S3-1	75	75 BAUD	* Do not turn more than one switch on at a time
S3-2	11	110 BAUD	
S3-3	15	150 BAUD	
S3-4	30	300 BAUD	
S3-5	60	600 BAUD	
S3-6	12	1200 BAUD	
S3-7	24/48	2400 or 4800 (normally 2400 if not jumpered K to M)	
S3-8	96	9600 BAUD	

# SERIAL I/O CONTROL -- Schematic Drawing #3

Switch No.	Mnemonic	ON	OFF
S4-1	PS	Parity even	Parity odd (if S4-5 on)
S4-2	WLS 1	Data word length	8bits 7bits 6bits 5bits
S4-3	WLS 2		Off Off On On
			Off On Off On
S4-4	SBS	1 stop bit	2 stop bits (1.5 if 5bits/word)
S4-5	PI	Parity	No parity
S4-6	F/H	Half duplex	Full duplex

## MEMORY ALLOCATION: ON CARD

Hexidecimal Address	Function
C000 - C7FF	Personality Module ROM or PROM (2048 words)
C800 - CBFF	System RAM (1024 words)
CC00 - CFFF	Display RAM Memory (1024 characters)

## ON CARD INPUT PORT ALLOCATION

Hexidecimal Port Address	Function
F8	Status, Serial Comm. channel
F9	Serial Communication Channel Data
FA	Aux. Status, Cassette tape interface, parallel I/O, keyboard input
FB	Audio Cassette (CUTS) Data
FC	Keyboard Data (from J3)
FD	Parallel Port Data (from J2)
FE	Display Status
FF	Sense Switch (S2-1 thru S2-8)

## OUTPUT PORTS

Hex Port Address	Function
F8	Control, Serial Comm. Channel
F9	Data, Serial Comm. Channel
FA	Control, Parallel I/O, CUTS Cassette I/O
FB	Data, CUTS audio cassette Interface
FC	Alarm (optional)
FD	Data, Parallel output Data channel
FE	Scroll control, Display Section
FF	not used in Sol-PC

Rev A

### STATUS PORT INPUT BIT ASSIGNMENTS

#### PORT F8 (STATUS, SERIAL COMM. CHANNEL)

<u>BIT</u>	<u>SIGNAL NAME</u>	<u>FUNCTION</u>	<u>ACTIVE DIRECTION</u>
Ø	SCD	Serial Carrier Detect (EIA)	1 carrier
1	SDSR	Serial Data Set Ready (EIA)	Ø link ok
2	SPE	Serial Parity Error	1 error
3	SFE	Serial Framing Error	1 error
4	SOE	Serial Overrun Error	1 error
5	SCTS	Serial Clear to Send (EIA)	Ø clear
6	SDR	UART Serial Data Ready	1 ready
7	STBE	UART Serial Transmit Buffer Empty	1 empty

#### PORT FA (AUX. STATUS, CASSETTE TAPE INTERFACE, PARALLEL I/O, KEYBOARD INPUT)

<u>BIT</u>	<u>SIGNAL NAME</u>	<u>FUNCTION</u>	<u>ACTIVE DIRECTION</u>
Ø	KDR	Keyboard Data Ready	Ø ready
1	PDR	Parallel Data Ready	Ø ready
2	PXDR	Parallel eXternal Device Ready	Ø ready
3	TFE	Tape Framing Error	1 error
4	TOE	Tape Overrun Error	1 error
5	not used		
6	TDR	Tape Data Ready	1 ready
7	TTBE	Tape Transmitter Buffer Empty	1 empty

#### PORT FE (DISPLAY STATUS)

<u>BIT</u>	<u>SIGNAL NAME</u>	<u>FUNCTION</u>	<u>ACTIVE DIRECTION</u>
Ø	SOK	Scroll OK; ¼ sec timeout after scroll	Ø time complete

### CONTROL PORT OUTPUT BIT ASSIGNMENTS

#### PORT F8 (CONTROL, SERIAL COMM. CHANNEL)

<u>BIT</u>	<u>SIGNAL NAME</u>	<u>FUNCTION</u>	<u>ACTIVE DIRECTION</u>
4	SRTS	Serial Request to Send	1 request

#### PORT FA (CONTROL, PARALLEL I/O, CUTS CASSETTE I/O)

<u>BIT</u>	<u>SIGNAL NAME</u>	<u>FUNCTION</u>	<u>ACTIVE DIRECTION</u>
3	PIE	Parallel Input Enable	1 pin 3 J2 low
4	PUS	Parallel Unit Select	0 pin 14 J2 low
5	TBR	Tape Baud Rate (300/1200)	0 1200 Baud
6	TT2	Tape Transport 2	0 run tape
7	TT1	Tape Transport 1	0 run tape

#### PORT FE (SCROLL CONTROL, DISPLAY SECTION)

<u>BIT</u>	<u>SIGNAL NAME</u>	<u>FUNCTION</u>	<u>ACTIVE DIRECTION</u>
Ø - 3	BDLA	Beginning Display Line Absolute address	4-bit data nybble
4 - 7	FDSP	First Displayed Line Screen Position	4-bit data nybble

### CONNECTOR DESIGNATION

J1	Serial data	J6	Cassette Tape Audio Out
J2	Parallel Data	J7	Cassette Tape Audio In
J3	Keyboard	J8	Tape Motor 1
J4	Display Expansion	J9	Tape Motor 2
J5	ROM Personality Module	J10	PC Power
		J11	S-100 Bus Expansion

# YOUR PERSONAL GENIE

by Tom Munnecke

It helps you with your income tax, then it takes you in the Starship Enterprise on an outer space crusade against the Klingons. It teaches you Boolean logic, then it becomes an opponent at checkers. It draws vivid pictures on your television set, then telephones a distant computer to calculate the value of your personal stock portfolio.

What is this personal genie? How can it take on so many personalities? It is the personal computer, and its personalities are the unique products of its programmer. The computer is capable of nothing more, nothing less than the programmer instructing it. For all the precision and rigidity associated with a computer, the programmer's work is still a uniquely personal reflection of himself.

The fundamental connection between the programmer and the computer is the computer language. The increasing number and sophistication of computer languages bring the power of the personal computer to the non-professional.

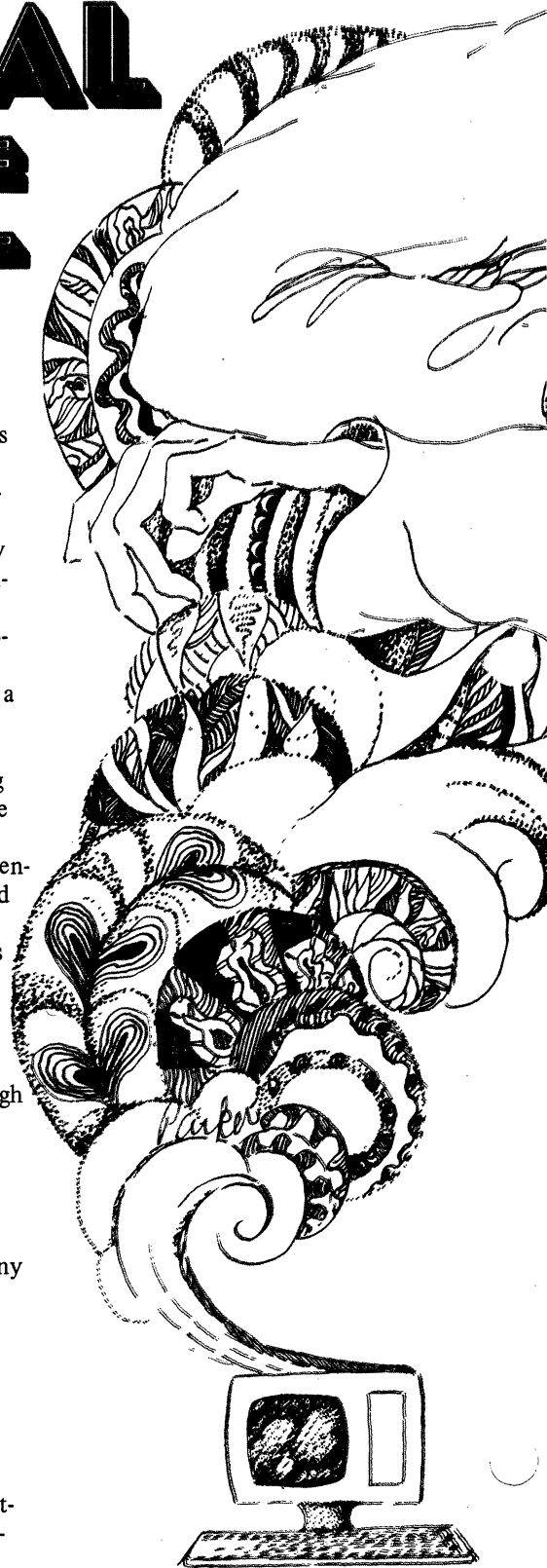
Computers are simple to deal with once certain fundamentals are understood. After that, learning becomes a trial and error experience. A person learning to walk does not need to understand each muscle, joint, and bone; he simply tries to walk and corrects his mistakes. So it is with computer programming. The novice programmer does not need to know the intricacies of the computer. He needs only: to know the fundamentals of the language, to know what his errors are and how to correct them, and to have time enough to try out his ideas.

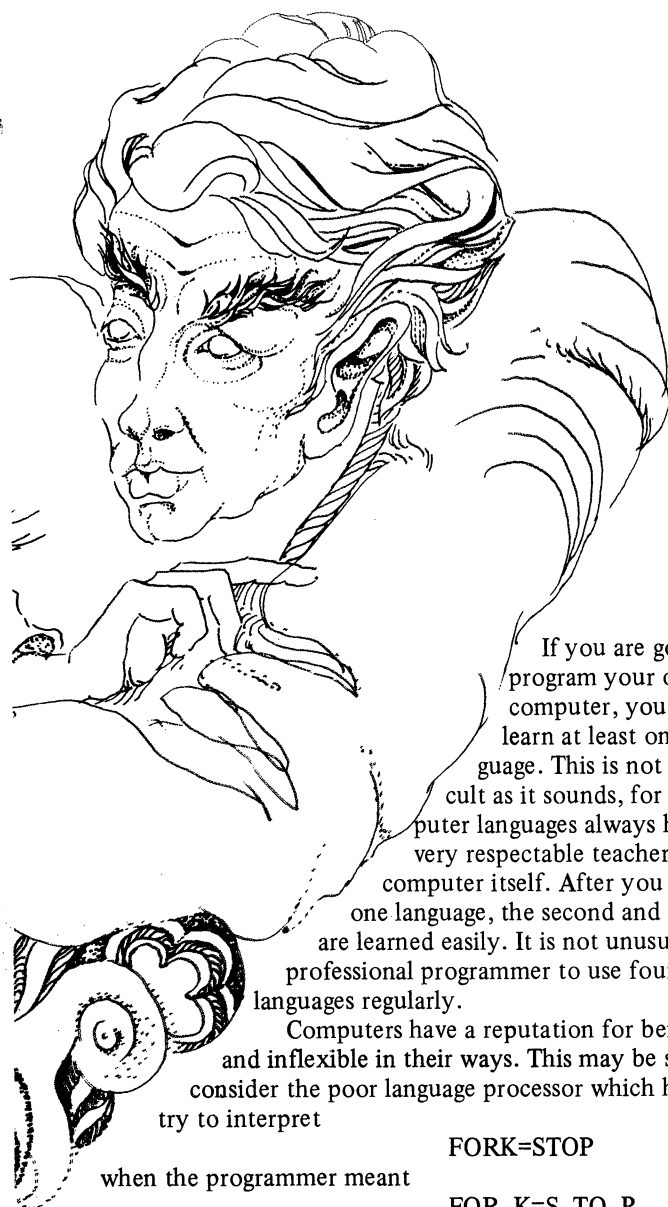
The personal computer is a tool — the most powerful tool ever put in the hands of the private individual. Its potential is limited only by its owner's capacity to apply it.

This article provides a head start on learning any computer language, discussing the merits and drawbacks of many of the computer languages available to the personal computing enthusiast.

## *What is a Computer Language?*

Computers operate in sequences of primitive decisions made in millionths of seconds. People think in terms of vague concepts derived over days and months. The computer language is the means of linking these vague human concepts to the primitive computer decision.





If you are going to program your own computer, you need to learn at least one language. This is not as difficult as it sounds, for computer languages always have a very respectable teacher — the computer itself. After you learn one language, the second and third are learned easily. It is not unusual for a professional programmer to use four or five languages regularly.

Computers have a reputation for being rigid and inflexible in their ways. This may be so, but consider the poor language processor which has to try to interpret

`FORK=STOP`

when the programmer meant

`FOR K=S TO P`

Most of the rigidity of the computer is there for a purpose. If you learn how they interpret things, some apparent inflexibility will fade away.

In order for the computer and the programmer to communicate, they must have some common physical medium for communicating. Usually, this is a keyboard/printer or video display. The programmer enters his programs in whatever language he is using, in his version of the language, known as the source language. He then asks a language processor to prepare it for the computer to process it.

There are two types of language processors — translators and interpreters. The translator accepts the source language and translates it to an object language, which is then loaded into the computer to be executed. Translators are further broken down into assemblers and compilers. The assembler is a means of manipulating machine-level operations for a specific computer, while the compiler translates higher-level, or more human-oriented languages. Interpreters execute the source language directly without the intermediate process of translating to an object language.

Languages are classified into two vaguely defined classifications: high-level and low-level. A low-level language is

one in which each of the source code instructions corresponds to a machine-level operation. Source code in a high-level language may generate many machine-level instructions.

### *Assemblers, Compilers, and Interpreters*

Each of the types of language processors has its merits and drawbacks — assemblers give the programmer great power but require very detailed instructions; compilers support higher-level languages, but sacrifice machine efficiency; and interpreters are easy to use, but are not as efficient as compilers.

### *Assemblers*

The assembler is the simplest form of computer language. It accepts source code and translates it one-for-one into machine-level instructions or object code. Thus, the programmer has detailed control (and responsibility) of each instruction. For example, the programmer might write a line in assembler such as:

`NEXT JSR INCHAR ; Jump to subroutine to get a character.`

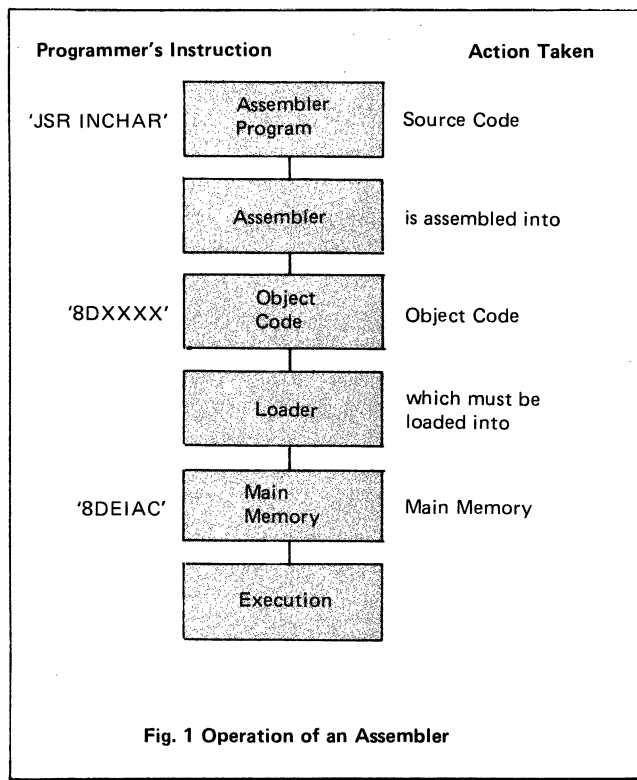
'NEXT' is a label for the line. 'JSR' is a mnemonic for the Motorola 6800 instruction 'Jump to subroutine'. 'IN-CHAR' represents the address of the subroutine to be used. 'Jump . . .' is a comment inserted by the programmer to explain the instruction for documentation.

The assembler (for the 6800) will assemble this instruction into the hexadecimal '8DXXXX' where '8D' is the operation code for branch to subroutine, and 'XXXX' is the address of subroutine INCHAR. See Fig. 1.

Since the assembler may not know where the INCHAR subroutine is to be located when the program is executed, it must be resolved at a later time by the *loader* program.

### *Compilers*

The compiler acts much like the assembler, but works with higher level languages. The compiler understands more



complex expressions, and does much more work than the assembler. Figure 2 illustrates a single high level language expression which would require 6 lines to write in assembler.

Compiler	Assembler
TOT=SUM+NUM	LDA A SUM+1    Add the right ADD A NUM+1    most bytes and STA A TOT+1    store the result. LDA A SUM       Add the left ADC A NUM       most bytes and STA A TOT       store the result.

**Fig. 2 Comparison of high-level expression and its low-level language equivalent.**

This is a simple example, but a more complex example, such as:

$TOT = (SUM + NUM / 1.238 * \cos(ARC/360)) ** 2 / 7.32$

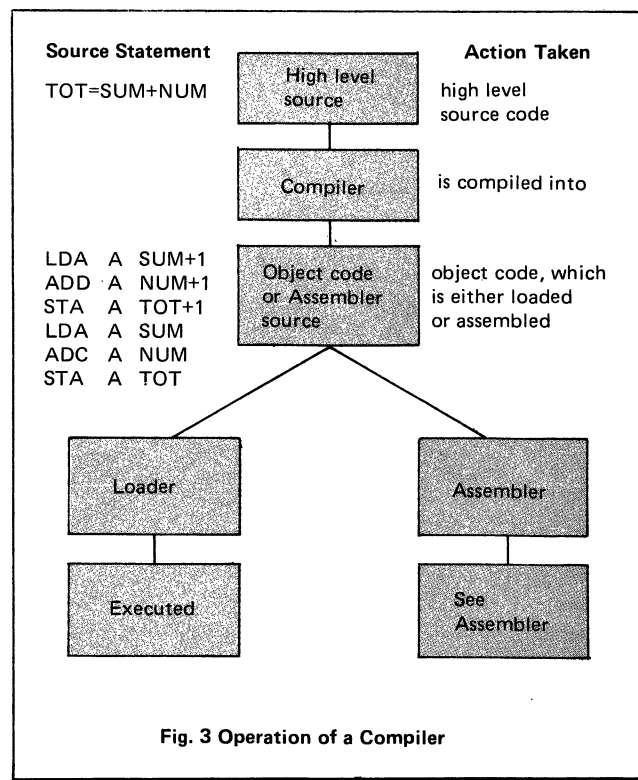
could give the assembler language programmer a tremendous amount of difficulty.

Typically, a compiler produces assembly language code, which is then passed through the assembler.

### Interpreters

The interpreter is a departure from the techniques of the assembler or compiler. While the translators create a program which must be loaded and executed later, the interpreter executes source instructions directly. The source remains in its original form.

Many languages may be either compiled or interpreted, although some features of a language may make compilation difficult, if not impossible. The interpreted language can change its interpretation as it receives new data, while the compiler does not know what data the program will receive until after it has finished its work.



### Comparison of the Methods:

Each of the methods is used in the commercial computing world, indicating that there is sound economic need for each. The methods may intermingle, as in compilers that accept assembler language code, incremental compilers, which are a cross between interpreting and compiling, and compilers which produce interpretive object code. Fig. 5 illustrates many common considerations of the various language processors.

### Disadvantages of Assemblers

Because the programmer must detail each operation of the computer, his workload is much greater than with higher level languages. His chances for making an error are much greater than in high-level languages. The programmer can easily become enmeshed in the maze of details he must remember. Modifying an intricate assembler language program may be very difficult, if not impossible. Assemblers are not usually interactive, requiring the entire program to be re-assembled when an error is made.

### Advantages of Compilers

The compiler is capable of supporting much higher level languages than assembler or macro assembler. The programmer can work faster, make fewer errors, and learn the language faster than he can assembler. The compiler's object code may be executed much faster than an interpreter could execute the program (between 5 and 10 times faster). Programs written in the higher level language may be recompiled on a new type of computer, without modifying the program.

### Disadvantages of Compilers

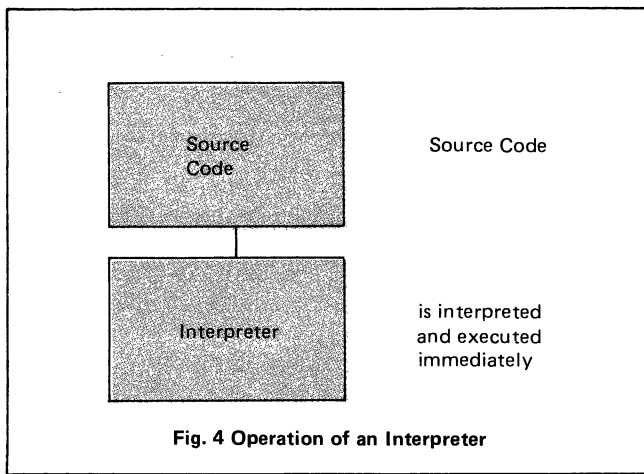
Compilers are usually large, complex programs which require some time to compile a program, in addition to a significant amount of off-line storage. Compilers are not

usually interactive, because they require an entire program to be recompiled when a single change must be made.

Due to the internal workings of the compiler, data types must be fixed during compilation. This process, known as binding, reduces the program's ability to adapt to new data as the program is executed. An interpreter, however, does not bind its variables until execution.

#### Advantages of the Interpreter

Since the interpreter executes its source code directly, the programmer may interact more directly with the computer. Usually, the interpreter provides a direct mode, where the programmer may execute statements directly as he enters them, and an indirect mode, where his commands are stored in a program for later execution. The programmer can usually stop the program, examine variables, and resume execution. Some interpreters (such as APL and MUMPS) provide an EXECUTE command, which allows the program to execute a character string as if it were program text. Conversely, some interpreters (MUMPS) allow a program to treat its own text as data. Interpreters are useful for systems where the language processor needs to be 'built in' to the computer, as in intelligent terminals.



Characteristic	Compiler	Interpreter	Assembler
Binding Time	Compile	Execution	Assembly
Off line Storage	much	little/none	much
CPU Efficiency	medium	low	high
Programmer Efficiency	medium	high	low
Program Size	large	medium	small
Error Detection Language	machine	source	machine
Interactive Debugging	no	yes	no
Language Processing Efficiency	low	medium	high

**Fig 5 Comparison of the features of the various types of language processors.**

- Binding time — the point when the program's data types are fixed.
- Offline Storage — the amount of storage such as floppy disks, cassettes, etc, required for the language processor to work.
- CPU Efficiency — of the program being processed.
- Programmer Efficiency — of the programmer writing the program to be processed.
- Program Size — of the object code, or source code, in the case of the interpreter.
- Error Detection Language — the language in which run time errors are detected.

#### Disadvantages of the Interpreter

Interpreters tend to be slower than compilers, between 5 and 10 times slower, as a rule of thumb. This slowness is due to the interpreter's need to analyze each statement every time it sees it, whereas the compiler need analyze it only once. The interpreter program must remain in memory for even a small program.

#### A Bit of History

The first computers were large, expensive devices requiring a roomful of air conditioning just to keep them cool. Programming them was very difficult, and they ran quite slow:

"... the machine will then continue in operation hour after hour, completely checking its own results until either the problem is solved, or a breakdown occurs" (A Manual of Operation for the Automated Sequence Controlled Calculator, Harvard University, 1946).

At that time, a computer cost millions of dollars, and a programmer cost a few hundred dollars per month. Today, a computer costs hundreds of dollars, and the programmer costs thousands of dollars per month. To put it in another way, in 1946 a computer cost the equivalent of 250 programmers, today the programmer costs the equivalent of 100 computers.

Everyone agrees that computers should be used 'efficiently'. The problem is that people think of making the CPU efficient, not the person using it. The microcomputer has undermined the conventional wisdom of computer efficiency. The person who spends several month's rent on a personal computer wants to see it do something for him immediately, regardless of whether it uses the CPU 'efficiently'. Chances are he uses the computer only a few hours a day. On the other hand, the professional programmer who works as one of a score of programmers using a large computer must contend with CPU efficiency in order to keep from overloading the computer.

The microcomputer user needs to worry about CPU efficiency only when he reaches some limit — not enough memory response not fast enough, etc. Since no one else is waiting to use his computer, he does not have to worry about inefficiencies which do not force him beyond his limits.

The large computer programmer, however, must constantly worry about sharing the computer with all the other users. Even if a program works fast enough for him, and uses little enough memory, it still must be made 'efficient' for the other users of the system.

As a result of this historical concern for CPU efficiency, people are fixated on "making the computer run efficiently". Language design has been heavily weighted in favor of making the computer efficient, not the programmer.

The personal computing software scene was a completely unforeseen turn of events. None of the language designers ever thought that the programmer would be working alone on his own computer. As a result, the design tradeoffs were heavily slanted in favor of the commercial user.

#### Which language is Best?

"I speak Spanish to God, Italian to women, French to men, and German to my horse".

Charles V of France.

What is the best language? BASIC? Assembler? PL/M,



<p>PROGRAM</p> <p>LINE</p> <p>LINE NUMBER COMMANDS ARGUMENTS</p> <p>Fig. 6 BASIC Program Elements</p>	<p>PROGRAM</p> <p>STATEMENTS</p> <p>LABEL OPERATION CODE OPERANDS</p> <p>Fig. 7 Assembler Program Elements</p>	<p>PROGRAM</p> <p>BLOCK</p> <p>STATEMENTS EXPRESSIONS</p> <p>Fig. 8 PL/M Program Elements</p>
<p>PROGRAM</p> <p>ROUTINE</p> <p>LINE LABEL COMMAND ARGUMENTS</p> <p>Fig. 9 MUMPS Program Elements</p>	<p>WORKSPACE</p> <p>FUNCTION</p> <p>LINE</p> <p>OPERATORS LITERALS FUNCTION REFERENCES</p> <p>Fig. 10 APL Program Elements</p>	<p>PROGRAM</p> <p>GROUP</p> <p>LINE NUMBER COMMAND ARGUMENTS</p> <p>Fig. 11 FOCAL Program Elements</p>

MUMPS, APL, PASCAL, FORTRAN, SNOBOL, COBOL, LISP, COMIT, MAD, or any of the hundreds of others? And after the best language is chosen, which dialect is best? Consider the dialects of BASIC: Tiny BASIC, Extended BASIC, BASIC Plus, Business BASIC, ANS BASIC . . .

Perhaps a good analogy could be drawn between computer languages and spoken languages. Which spoken language is best? English? French? Chinese? Italian? It all depends on what you want to do with it. If you are in Paris, French would be a good contender for the 'best' language. Suppose you are in Kansas, and believed Charles' statement above that Italian is best for speaking to women. Romantic pretensions aside, you would probably have better luck with English.

The "best" computer language is not selected on the basis of its syntax or grammar. It is a very pragmatic decision based on what is available, what the programmer knows, whether it can perform the task at hand, and what programs are available to him from other sources.

The selection of a computer language is an important decision to the personal programmer for many reasons beyond the above pragmatic ones. The language a programmer uses profoundly affects the way he sees a problem. As Whorf said, "We dissect nature along lines laid down by our native language". The APL programmer thinks in terms of vectors, the MUMPS programmer thinks in terms of data bases, and the Assembly language programmer thinks in terms of individual bytes of memory.

Therefore, in reviewing each of the languages, the reader must apply them to his own needs. The following list is a sample of some of the languages available (or may be soon) to the micro-computer user.

**BASIC** — (Beginner's All purpose Symbolic Instruction Code). This is the most common high-level language used on personal computers. It is a very simple, easy to learn language. There is a large library of programs available,

since BASIC is used in many universities and schools. Because it is a simple language, it is somewhat limited and difficult to use for some complex problems. BASIC is usually interpreted on microcomputers, although some compilers exist. Programs written in BASIC for one computer can often be run on another with only slight changes.

**Assembler** — Assembler language is commonly used on personal computers. Since many personal computers have neither the memory or Input/Output capability to run an assembler, the programmer often manually assembles his program and enters it through the switches on the panel. Assembler language is unique to each computer, so program exchange is limited to one particular computer type.

Assembler language is the common denominator of all programs — eventually, all programs are just a sequence of assembler-level instructions. Therefore, any one wishing to really know how his computer works must learn at least a little Assembler. Often, a program is written in a high-level language which calls an Assembler language subroutine for difficult or critical portions of logic. This can be a very economical mix for programs which exceed the limits of a high-level language.

**PL/M** — (A program name copyrighted by Intel Corp.) is a compiled language derived from IBM's PL/1. Versions exist for the 8080, 6800, and Signetics 2650. Some high speed, mass storage (floppy disk, for example) is required. It is an alternative to assembler, producing slightly less efficient programs in much less programming time. A basic user would find PL/M difficult to use for simple problems, but easier to use for more complex problems. There is no extensive library of programs in PL/M as with BASIC.

**MUMPS** — (Massachusetts General Hospital Utility Multi-Programming System) is an interpretive language oriented towards interactive data management applications. MUMPS has many characteristics of BASIC, FOCAL, and IBM's PL/1. It differs from all these in that it has built-in data base capabilities for handling data on mass storage devices. Although not widely available on microcomputers now, the National Bureau of Standards published a standard version (NBS Handbook 118) which details how one would write an interpreter for MUMPS.

MUMPS has extensive data handling capabilities, suited for applications such as personal accounting, word processing, and general information systems. Since the development of MUMPS was federally supported, much MUMPS software is in the public domain.

**APL** — (A Programming Language) is a computer language derived from Iverson's elegant mathematical notation. It is a very powerful mathematical tool, having primitive functions for matrix inversion, inner products, sorting, and many other areas. Although initially developed for large scale computers, it is now available for portable commercial computers. APL is usually interpreted, and therefore well suited for interactive personal computing.

**FOCAL** — (Formulating On-Line Calculations in Algebraic Language) is a language brought out as an early on-line language for calculations. Its syntax is similar to MUMPS, although its functions are closer to BASIC. FOCAL is available on the 8080 and has a modest programming library.

#### *Learning a Computer Language*

Your first task in learning a new language is to build up a basic understanding of the language. This can be gained from the reference manual for the language distributed with the software. Magazines such as Personal Computing carry many articles on the more popular languages. There is a variety of books available in libraries and computer stores, and more advertised in professional data processing magazines.

When studying a language, it is helpful to divide the project into three areas:

**SYNTAX** — How you say something

**SEMANTICS** — What you mean

**PRAGMATICS** — How you make the language do what you want

**Syntax.** The syntax of the language is usually the quickest part to learn. How does the language distinguish between a number and a variable? Do you need a number before each line? What characters are allowed by the language?

**Semantics.** The semantic aspects of the language are more difficult to learn, but you do not have to understand everything to use the language. What are arithmetic functions in the language? How do you retrieve data from the terminal? How do you format output?

**Pragmatics.** This is the most difficult portion to learn, yet it is the skill most easily carried over to other languages. How do you make the language solve your problem? How do you create, change, and delete programs? Can you stop the program while it is executing, examine the state of things, then resume execution?

These three classifications are very useful for comparing languages. For example, BASIC, FOCAL and FORTRAN have similar semantics but different syntaxes. MUMPS and

FOCAL have similar syntaxes, but different semantics.

With this background, you should be able to modify a simple program to make it do increasingly complex tasks. Each time you modify the program, use some new aspect of the language, being careful to add one aspect at a time. Then, try the new version to see if it does what you expect.

Each step of the way, you will be informed of your mistakes by your friendly adversary, the computer.

#### *The Importance of Making Errors*

"Nine times out of ten, in the arts as well as life, there is actually no truth to be discovered; there is only error to be exposed."

H.L. Mencken

Making an error in a computer program is a fundamental source of learning. You tried something and the computer told you it didn't work. The programmer who proudly announces "my last program worked the first time without any bugs" is a programmer who probably did not learn anything new writing it.

	BASIC	MUMPS	APL	FOCAL	PL/M	FORTRAN
Integer (16 Bit)	X	X	X	X	X	X
Byte			X		X	
Character String	X	X				
Floating Point	X	X	X	X		X
Logical		X	X			X
Labels		X				

Fig. 12 Cross Index of Data Element Types

	BASIC	MUMPS	APL	FOCAL	PL/M	FORTRAN
Assignment	LET	SET	←	SET	=	=
Read from Console	INPUT	READ	← □	ASK	INPUT	INPUT
Write to Console	PRINT	WRITE	□ ←	TYPE	OUTPUT	OUTPUT

Fig. 13 Cross Index of Data Movement

	BASIC	MUMPS	APL	FOCAL	PL/M	FORTRAN
Unconditional Branch	GOTO	GOTO	→	GO	GOTO	GOTO
Conditional Branching	IF	IF	→	IF	IF	IF
Invocation	GO SUB	DO	NAME	DO	CALL	CALL
Return from Invocation	RE-TURN	QUIT	→ 0	QUIT	END	RETURN
Looping	FOR/NEXT	FOR		FOR	DO	DO

Fig. 14 Cross Index of Control of Flow

BASIC MUMPS APL FOCAL PL/M FORTRAN					
And	&	^			
Or	!	✓			
Not	'	~			
Greater Than	>	>	*		
Less Than	<	<	*		
Equal	=	=	*		
Not Equal	<>		*		
Less Than Or Eq.	<=	≤	*		
Greater Than or Equal	=>	≥	*		

Fig. 15 Logical and Arithmetic Comparison Function

\*handled by IF statement structure.

BASIC MUMPS APL FOCAL PL/M					
Addition	+	+	+	+	+
Subtraction	-	-	-	-	-
Divide	/	/	÷	/	/
Multiply	*	*	X	*	*
Exponentiation	↑			FExP	
Square root	SQR		*.5	FSQT	
Cosine	COS		2°	FCOS	
Tangent	TAN		3°		
SINE	SIN		1°	FSIN	
e <sup>x</sup> Exponential	EXP		*	EXP	
Natural log	LOG		⊕	FLOG	
Absolute Val	ABS		!	FABS	
Greatest Integer	INT		⌊	FITR	
Random Number	RND	\$R	?	FRAN	
Signum	SGN		X	FSGN	
Modulo		#	!		

Fig. 16 Cross Index of Arithmetic Functions

The absence of an error when writing a program indicates only that a situation new to the programmer did not come up — not that the programmer has learned the language.

There are generally four types of errors: syntax, semantic, pragmatic, and covert.

### Syntax errors

The syntax error is the most common error which faces the beginning programmer. A syntax error is a statement that violates the language's basic rule for expression. Typically, they are caused by:

- typing errors — a finger slips to the wrong key, a zero instead of the letter O, etc.
- misunderstanding the syntax. The new programmer may not understand that he has to put a comma between variables in a print statement, or put apostrophes around literals.
- confusing the syntax. The programmer might confuse a colon and the comma, or, he might carry over some syntax from another language he knows.

One thing in common with all these errors is that the computer can detect them. In most interpreters, the programmer may directly enter and execute any questionable statements.

The lesson is clear: When in doubt, try it. Let the computer tell you whether it will accept the statement. Many manuals are not reliable enough to trust anyway.

The above advice flies directly in the face of conventional computer programming wisdom. In the past, there was considerable stigma attached to anyone found 'letting the computer do his debugging'. The theory was, that the computer is a valuable resource, and that a programmer should not waste computer time. Instead, he should carefully desk-check his program before each submission. In the microcomputer world, this philosophy is radically altered. It makes no sense for the programmer to check his work on paper when his computer is waiting for him to enter it.

### Semantic errors

These errors are also common in the early stages of learning a new language, but continue to plague the programmer throughout the use of the language. These errors are statements which are syntactically correct, but do not perform the function desired by the programmer. Some typical semantic errors are:

- Mode errors — the programmer tries to add a number to a character string, but the language does not handle the conversion.
- Binding errors — the programmer names the wrong variable, label or subroutine.
- Juxtaposition or sequencing errors. An end of a loop is placed too far down in the program, or a variable is used before it is initialized.

Most of the same advice for syntax errors applies to grammatical errors. Sometimes, grammatical errors can slip through and only be detected by erratic program behavior.

### Pragmatic errors

The pragmatic error is a statement which is syntactically and semantically correct, but does not do what the programmer wants it to. These cannot be caught by the language processor. Typical pragmatic errors are:

- wrong function or command — the programmer uses a sine function instead of cosine.
- an improper formula — the programmer thought that Interest was Principal divided by Rate instead of Principal times Rate.

Pragmatic errors tend to be the last errors in a program to be detected, if only because the programmer will not see them until he cleans up the syntax and semantic errors and the program executes.

Pragmatic errors can be very difficult to detect, particularly in programs which are time dependent or involve much

BASIC MUMPS APL FOCAL PL/M					
Search		\$FIND	?		
Extract	MID\$ LEFT\$ RIGHT\$	\$EX-TRACT	SUB-SCRIPTS		
Concatenation		-			
Convert String to Number		\$ASCII			
Convert Number to String		\$CHAR			
Length	LEN\$	\$LENGTH			

Fig. 17 Cross Index String Functions

logic. Pragmatic errors are generally discovered with what the computing world euphemistically calls "testing". "I'll test this program to make sure it won't blow up," is an often heard phrase. Unfortunately after he completes his testing, he all too often says "my program blew up".

Testing can confirm the existence of an error, not that one doesn't exist. Just because 99 combinations of input data were tried does not guarantee that the hundreth combination will not fail.

#### Covert errors

When a program is tested and declared correct by the programmer, any remaining errors are by definition covert. These are insidious problems that appear only when events combine to form some previously untried condition. Some covert errors are:

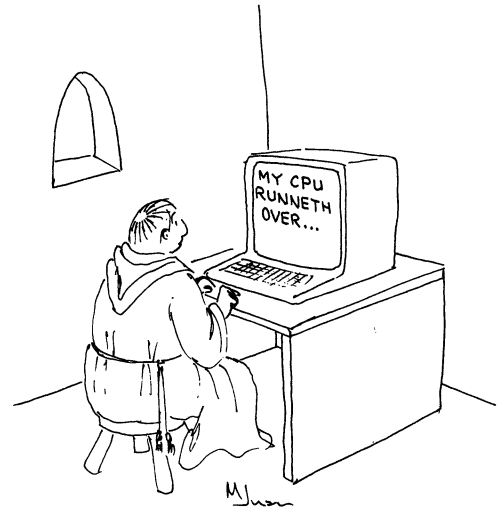
- a) An angle in a trigonometric equation goes to zero, causing a zero divide error in a later division.
- b) Improper data is entered, which the program does not reject as invalid. Recently, a program sent out a letter to the Emmet County Jail, "Dear Emmet C. Jail, you are among a select group of persons . . ." As the saying goes — garbage in, garbage out.
- c) The programmer leaves room for only 3 digits of a number, but the number grows past 999.

Covert errors always have and always will exist in computer software. However, a great deal of attention in computer science circles has been given to writing programs which may be "proved" correct. These efforts, named "structured programming", "software engineering", and "composite design" will be covered in a future article. The

fundamental principles common to these are:

- a) Break the big problem into clusters of independent little problems.
- b) Link the clusters together in a hierarchical manner such that each cluster is independently testable.
- c) Limit the number of paths the program may take. This is accomplished by limiting the use of the GOTO statement.

The programmer should learn to improve his skills by analyzing the errors he makes. When he meets that benevolent dictator of linguistic purity — the error message — he should treat it as a means of learning a little more about the language.



## UPDATES

### Sol TERMINAL COMPUTER<sup>TM</sup>

Electronics is a very fast moving field. Development of new products, and improvements in the old products proceeds at an unprecedented rate. The continuing development of the Sol Terminal Computer is no exception. Better parts become available and are included, experience yields circuit improvements, and new circuitry is developed. This process generates changes much more frequently than this manual is reprinted. As a result, we include the improvements as blue update sheets, added to this section as they become available. Be sure to integrate this information into the body of the manual before beginning, by making indicated changes in the text, adding or replacing pages, or making notes referring you to the update page.

If you have a question as to the currency of a particular page of text, look in the lower left-hand corner of the page. The initial version of the page will have this corner blank. When the contents of the page have changed, the new version will have "Rev A" in this corner; a third version will have "Rev B", and so forth. When a whole new page and page number are added, the corner is blank.

## Sol MANUAL - CHANGE NOTICE

### CHANGE NOTICE #11

Refer to Section X, Drawing X-19, Sol Audio Tape I/O Schematic. U110, a 4046 phase-locked-loop IC, has its VCO center frequency adjusted by VR3, shown as a 50K potentiometer. The upper frequency limit of adjustment is determined by R154, shown as a 100K resistor. The lower frequency limit is determined by the total resistance of VR3 and R154. Due to extreme variations in the specifications of this part among various vendors, the range of adjustment provided by VR3 and R154 is occasionally insufficient. To correct this problem, the value of VR3 has been changed to 100K, and the value of R154 has been changed to 47K. Parts of these new values have been included in your kit or assembled Sol. To insure that the parts are correctly installed if you have a kit, and in any case to insure that the manual reflects these changes, make the following notes in the manual:

Item	Page No.	Figure or Step No.	Changes
1	X-19	Schematic	Change VR3 value to 100K
2	"	"	Change R154 value to 47K
3	X-3	Sol PC Assembly	Change VR3 value to 100K
4	"	"	Change R154 value to 47K
5	III-3	Table 3-1.	Change Qty 47K resistor to 2
6	"	"	Change Qty 50K pot. to 2
7	"	"	Add 1 100K pot.
8	"	"	Change Qty 100K resistor to 3
9	III-36	Step 60	Change R154 value to 47K, yellow-violet-orange; VR3-100K
10	III-38	Step 70	Add note: "See Change Notice #11."
11	III-7	Section 3.4	Under item 11, Oscilloscope, delete "(optional)", add "with calibrated time base".

Make the following additional changes in the manual, unrelated to the substitution of new values for VR3 and R154:

11	III-24	Step 28	After sub-step relating to Figure 3-8, add note: "Adjust VR1 and VR2 for centering of the display."
12	"	"	In last sub-step on this page, change U49 to U59.



<u>Item</u>	<u>Page No.</u>	<u>Figure or Step No.</u>	<u>Changes</u>
13	III-25	Step 28	Add note before Step 29, "Do Step 73 (p III-39) now."
14	III-27	Step 35	Add note to install: ( ) U93 74LS175, and ( ) U106 74LS175
15	III-30	Step 41	Delete installation of: ( ) U93 74LS175, and ( ) U106 74LS175
16	III-39	Step 71	Add note: "Mike input not recommended".
17	III-28	Step 38	In the third sub-step, add: "as in Figure 3-9".

In item 10 above, you made a note referring you to this Change Notice. Instead of the adjustment procedure given in Step 70 on page III-38, use the following procedure:

- ( ) Ground the "Audio In" jack J7 on Sol-PC.
- ( ) Apply power to Sol-PC.
- ( ) Using a high-impedance probe from an oscilloscope with a calibrated time base (a frequency counter is preferred, if available), monitor the VCO frequency appearing at pin 4 of U110 (type 4046).
- ( ) Adjust VR3 for a measured frequency of 14.0 kHz. This is a period of 71.4 usec.

Due to variations in the availability of ICs from various suppliers, a number of substitutions may be made of equivalent IC types. Please make the following changes in the manual on the pages given to reflect these possible substitutions:

<u>Item</u>	<u>Page Nos.</u>	<u>ICs</u>	<u>Main Type</u>	<u>Additional Substitutes</u>
1	III-2, III-35	U95,6	74173	8T10
2	III-2, III-13	U104	AM0026	MH0026, 0026, "xx"0026
3	III-2, III-28 III-38	U51, U69	TMS6011NC	S1883, AV-5-1013, TR1602B
4	V-2, V-7	U1, U2	74LS175	25LS175
5	V-2, V-7	U18	8574	many possible equivalents

Sol MANUAL ERRATA NOTICE #3

1. Reference Section X, Drawings, Drawing X-17, Input/Output.

In the Baud Rate Generator section of this schematic, the function of switch S3-7 is incorrectly shown as selecting 2400/4800 Baud, and S3-8 is incorrectly shown as selecting only 9600 Baud. Change the schematic to show that S3-7 selects 2400 Baud only, and that S3-8 selects 9600/4800 Baud. Draw a line connecting points L and M to indicate a jumper.

2. Reference Section VII, page VII-15, Table 7-2.

In the Baud Rate column of this table, change "4800\*\*\*" to read "9600\*\*\*". Also, in the footnote with the triple asterisk, change the phrase "SDI operates at 9600 Baud..." to read "SDI operates at 4800 Baud..."

## IMSAI 8080 SYSTEM BUS STRUCTURE

The IMSAI 8080 system bus structure consists of 100 lines. These are arranged 50 on each side of the plug-in boards, with pins 1 through 50 on the component side and pins 51 through 100 on the back side. As the board is viewed right-side up (components up, 100 pin connector towards you) pin #1 is on the left end on the top and pin 51 is on the back side directly opposite pin #1.

### Conventions:

SYMBOLS: "P" prefix indicates a processor command or control signal

"S" prefix indicates a processor status signal

LOADING: All inputs to a card should be loaded with a maximum of 1 TTL low power load

LEVELS: All bus signals except the power supply are TTL. All Data and Address lines are positive TRUE (ground = 0 bit)

### BUS DEFINITION

<u>Front Side</u>			
<u>No.</u>	<u>SYMBOL</u>	<u>NAME</u>	<u>FUNCTION</u>
1	+8V	+8 volts	Unregulated input to 5v regulators
2	+16V	+16 volts	Positive unregulated voltage
3	XRDY	External Ready	Used by Front Panel: Pulling this line low will cause the processor to enter a WAIT state and allows the status of the normal Ready Line (PRDY) to be examined.
4	V10	Vectored Interrupt Line #0	

BUS DEFINITION

<u>Front Side</u>				
<u>No.</u>	<u>SYMBOL</u>	<u>NAME</u>	<u>FUNCTION</u>	
5	V11	Vectored Interrupt Line # 1		
6	V12	Vectored Interrupt Line #2		
7	V13	Vectored Interrupt Line #3		
8	V14	Vectored Interrupt Line #4		
9	V15	Vectored Interrupt Line #5		
10	V16	Vectored Interrupt Line #6		
11	V17	Vectored Interrupt Line #7		
12 to 17	UNUSED			
18	<u>STATUS DSBL</u>	STATUS DISABLE	Allows the buffers for the 8 status lines to be tri- stated	
19	<u>CC DSB</u>	COMMAND CONTROL DISABLE	Allows the buffers for the 6 output command/control lines to be tri-stated	
20	UNPROT	UNPROTECT	Reserved for input to the memory pro- tect flip-flop on a given memory board	
21	SS	SINGLE STEP	Used by Front Panel to disable input buf- fer while panel drives bidirectional data bus	

# BUS DEFINITION

## Front Side

<u>No.</u>	<u>SYMBOL</u>	<u>NAME</u>	<u>FUNCTION</u>
22	<u>ADDR DSBL</u>	ADDRESS DISABLE	Allows the buffers for the 16 address lines to be tri-stated
23	<u>DO DSBL</u>	DATA OUT DISABLE	Allows the bidirectional data bus drivers for the 8 data lines to be tri-stated for both input and output data buses
24	ø2	Phase 2 Clock	
25	ø1	Phase 1 Clock	
26	PHLDA	Hold Acknowledge	Processor control output signal which appears in response to the HOLD signal; indicates that the data and address bus will go to the high impedance state on the 8080. Note: <u>ADDR DSBL and DO DSBL must be driven to tri-state the system bus</u>
27	PWAIT	WAIT	Processor control output signal which acknowledges that the processor is in a WAIT state
28	PINTE	INTERRUPT ENABLE	Processor control output signal indicating interrupts are enabled: may be set or reset by EI and DI instruction and inhibits interrupts from being accepted by the CPU if it is reset

# BUS DEFINITION

Front Side

<u>No.</u>	<u>SYMBOL</u>	<u>NAME</u>	<u>FUNCTION</u>
29	A5	Address Line #5	
30	A4	Address Line #4	
31	A3	Address Line #3	
32	A15	Address Line #15	
33	A12	Address Line #12	
34	A9	Address Line #9	
35	DO	Data Out Line #1	
36	DO0	Data Out Line #0	
37	A10	Address Line #10	
38	DO4	Data Out Line #4	
39	DO5	Data Out Line #5	
40	DO6	Data Out Line #6	
41	D12	Data In Line #2	
42	D13	Data In Line #3	
43	D17	Data In Line #7	
44	SM1	M1	Status output signal that indicates that the processor is in the fetch cycle for the first byte of an instruction
45	SOUT	OUT	Status output signal which indicates that the address bus contains the address of an output device and the data bus will contain the output data when <u>PWR</u> is active

## BUS DEFINITION

### Front Side

<u>No.</u>	<u>SYMBOL</u>	<u>NAME</u>	<u>FUNCTION</u>
46	SNIP	INP	Status output signal which indicates that that the address bus contains the address of an input device and the input data should be placed on the data bus when PDBIN is active
47	SMEMR	MEMR	Status output signal which indicates that the data bus will be used for memory read data
48	SHLTA	HLTA	Status output signal which acknowledges a HALT instruction
49	<u>CLOCK</u>	CLOCK	2 MHz clock signal
50	GND	GROUND	

### Back Side

<u>No.</u>	<u>SYMBOL</u>	<u>NAME</u>	<u>FUNCTION</u>
51	+8V	+8 volts	Unregulated input to 5v regulators
52	-16V	-16 volts	Negative unregulated voltage
53	<u>SSW DSB</u>	SENSE SWITCH DISABLE	Disables the data input buffers so the input from the sense switches may be strobed onto the bi-directional data bus
54	<u>EXT CLR</u>	EXTERNAL CLEAR	Clear signal for I/O devices (front panel switch closure to ground)



## BUS DEFINITION

## Back Side

<u>No.</u>	<u>SYMBOL</u>	<u>NAME</u>	<u>FUNCTION</u>
55	CGND	CHASSIS GROUND	
56 to 67	UNUSED		
68	MWRT	MEMORY WRITE	From the <u>Front Panel</u> indicates that the current data on the Data Out Bus is to be written into the memory location currently on the address bus
69	$\overline{\text{PS}}$	PROTECT STATUS	Reserved to indicate the status of the memory protect flip-flop on the memory board currently addressed
70	PROT	PROTECT	Reserved for input to the memory protect flip-flop on the memory board currently addressed
71	RUN	RUN	Indicates that the RUN/STOP flip-flop is set to run on the front panel
72	PRDY	READY	Processor command/control input that controls the run state of the processor; if the line is pulled low the processor will enter a wait state until the line is released

BUS DEFINITION

Back Side

<u>No.</u>	<u>SYMBOL</u>	<u>NAME</u>	<u>FUNCTION</u>
73	<u>PINT</u>	INTERRUPT REQUEST	The processor recognizes an interrupt request on this line at the end of the current instruction or while halted. If the processor is in the HOLD state or the Interrupt Enable flip-flop is reset, it will not honor the request
74	<u>PHOLD</u>	HOLD	Processor command input signal which requests the processor to enter the HOLD state; allows an external device to gain control of address and data buses as soon as the processor has completed its use of these buses for the current machine cycle
75	<u>PRESET</u>	RESET	Processor command input; while activated the content of the program counter is cleared and the instruction register is set to 0
76	PSYNC	SYNC	Processor control output provides a signal to indicate the beginning of each machine cycle
77	<u>PWR</u>	WRITE	Processor control output used for memory write or I/O output control; continued next page.

# BUS DEFINITION

Back Side

<u>No.</u>	<u>SYMBOL</u>	<u>NAME</u>	<u>FUNCTION</u>
77	<u>PWR</u>	WRITE	Con't.: data on the data bus is <u>stable</u> while the <u>PWR</u> is active
78	PDBIN	DATA BUS IN	Processor control output signal indicates to external circuits that the data bus is in the input mode
79	A0	Address Line #0	
80	A1	Address Line #1	
81	A2	Address Line #2	
82	A6	Address Line #6	
83	A7	Address Line #7	
84	A8	Address Line #8	
85	A13	Address Line #13	
86	A14	Address Line #14	
87	A11	Address Line #11	
88	DO2	Data Out Line #2	
89	DO3	Data Out Line #3	
90	DO7	Data Out Line #7	
91	D14	Data In Line #4	
92	D15	Data In Line #5	
93	D16	Data In Line #6	
94	D17	Data In Line #1	
95	D10	Data In Line #0	

# BUS DEFINITION

## Back Side

<u>No.</u>	<u>SYMBOL</u>	<u>NAME</u>	<u>FUNCTION</u>
96	SINTA	INTA	Status output signal to acknowledge signal for INTERRUPT request
97	SWO	WO	Status output signal indicates that the operation in the current machine cycle will be a WRITE memory or output function
98	SSTACK	STACK	Status output signal indicates that the address bus holds the pushdown stack address from the Stack Pointer
99	$\overline{\text{POC}}$	Power-On Clear	
100	GND	GROUND	

## THEORY OF OPERATIONS

The CP-A front panel assembly provides machine status indicators, user controlled switches, and control functions to the IMSAI 8080 operator. The CP-A board communicates with the MPU-A microprocessor and other boards through the 8080 back plane and, additionally, connects (via 16 conductor flat cable) to the bi-directional data bus of the 8080 microprocessor.

The CP-A panel uses 44 Light Emitting Diodes as front panel indicators. Many of these indicators directly correspond to signal levels on the IMSAI 8080 back plane, and are driven directly from the bus with no intervening logic. Indicators in this group are the 16 Address Bus LED's, the 8 STATUS byte LED's, the INTERRUPT ENABLED LED, the WAIT LED, and the HOLD LED.

The 8080 microprocessor chips bi-directional data bus levels (provided by a 16-conductor cable) are displayed on the DATA bus indicators via the 74LS04 (low power schottky hex inverter) sections. Also driven from the bi-directional bus is the 8212 8 bit latch used to drive the PROGRAMMED OUTPUT indicators. The RUN indicator is driven directly from the run/stop flip-flop (74107) on the CP-A Board.

The 16 ADDRESS-PROGRAMMED INPUT and ADDRESS-DATA switches allow the operator to place desired value (program, data, addresses) on the 8080's bi-directional bus.

As shown on the schematic, these switches connect 7405 (open collector) inverters to the bus in a wired- AND configuration.

Pullup resistors on the MPU Board ensure that the bus levels are all high unless any inverter on any one of the bus lines goes low. Thus, if an inverter goes low, (this condition will be discussed shortly) the address switch can be used to put either a high or low value on that line.

The function switches provide the operator with direct control of the microprocessor. The RUN/STOP switch controls the X-READY line via the RUN/STOP flip-flop.

If the switch is set to RUN, on the next falling edge of the Phase II clock, the RUN and X-READY lines are set high. If the switch is set to STOP, the high STOP value and the Phase II clock are Nanded (U16) and this value Nanded with the DATA OUT 5 bit (fetch/status) and the PROCESSOR SYNC line.

Thus, when the processor is fetching a new instruction, the RUN/STOP flip-flop will be reset, the processor X-READY line goes low, and the processor stops.

Several CP-A function switches operate by providing the 8080 with an instruction, executing the instruction, and then stopping the processor on the next cycle. The open collector 7405's and support gating put these instructions on the 8080's bi-directional bus. The EXAMINE function uses a jump instruction (hex C3) followed by two bytes of the address selected on the front panel switches.

This operation causes the processor to jump to the selected address and, then, the processor is stopped during the next cycle. When stopped, the processor was reading the selected byte from memory as if it were going to execute it. Therefore, the processor stops with the desired address displayed on the address bus and the contents of that address is displayed on the data bus.

If the RUN switch is operated at this time, the processor will continue to pull the selected byte from memory and execute it.

The EXAMINE NEXT and DEPOSIT NEXT switches use similar schemes and the NO-OP (hex 00 or octal 000) instruction to increment the address.

Much of the remaining logic of the CP-A is used to sequence these commands to provide the desired functions. The RUN/STOP flip-flop line, the SINGLE STEP line, the EXAMINE line, and the EXAMINE NEXT line are all input to an OR-gate controlling the X-READY line. (The X-READY line must be high for the processor to run. Its

function is identical to the P-READY line used by the memory and I/O boards. The X-READY line is reserved for use of the front panel to avoid conflicts of two gates driving the same backplane line). During each of these functions, the processor is permitted to execute an instruction, and then is stopped in the next cycle in a manner similar to the RUN/STOP flip-flop cycle described earlier.

For the SINGLE STEP function, a one-shot, triggered by the SINGLE-STEP switch, is used to produce a pulse and the trailing edge of that pulse is used to set a flip-flop which controls the SINGLE STEP line. This permits the processor to execute the present instruction. The SINGLE STEP flip-flop is reset by the occurrence of the sync pulse on the following instruction, thus causing the SINGLE STEP level to be removed and the processor to stop on the following cycle.

The EXAMINE-NEXT flip-flop is similarly controlled by the leading edge of a pulse from a one-shot driven by either the DEPOSIT NEXT or EXAMINE NEXT switch. The output of the flip-flop is used both to put the NO-OP (hex 00 or octal 000) onto the bi-directional data bus, and also to provide the READY signal so that the processor will execute the instruction. It is reset by the sync pulse on the following cycle, thus stopping the processor again.

The EXAMINE function involves a 4-step sequence produced by two flip-flops arranged as a counter. The pulse produced by the EXAMINE switch's one-shot starts the counter and on the first count, the jump instruction is inserted on the data bus. On successive counts of the two bit counter, the lower and upper address byte are inserted on the data bus in turn, and on the 4th count (that is, when the counter is back to 0), the processor is again stopped by the removal of the READY line. Thus, the EXAMINE logic provides the processor with the jump instruction and the two address bytes that the processor expects after a jump instruction and



stops the processor during the fetch of the designated memory byte.

Similarly, the DEPOSIT switch, when operated, produces a pulse from the DEPOSIT one-shot which is buffered to the MEMORY WRITE line on the backplane. The leading edge of this pulse also starts a second one-shot with a much longer period which puts the data from the data switches on to the data bus for the duration of the longer pulse. The DEPOSIT one-shots are triggered either by the operation of the DEPOSIT switch or by the trailing edge of the DEPOSIT NEXT one-shot so that the DEPOSIT function will operate at the end of the EXAMINE NEXT cycle.

The 7427 gate in U15.5 is used to insure that during the time the front panel is inserting any information on the bi-directional data bus, the MPU-A board's bi-directional data bus driver is not also trying to drive the bus at the same time.

The inputs to this gate are the DATA-ON line, the EXAMINE NEXT line and the EXAMINE line. These are the three functions during which the front panel is transferring data or instructions to the bus.

The inputs to the 7405 open-collector inverter bus drivers are the lines NO-OP, C3, HAD, and LAD. These levels are ANDed with the PDBIN signal so that the information appears on the bus during the time the processor is expecting to see it there.

The input port from the high order address switches is implemented simply by decoding the address FF and ANDing it with the DBIN signal so that switch values appear on the data bus during the time that the processor is expecting information from the port FF.

The same address decode signal is ANDed with the STATUS OUT line to enable the 8212 8 bit latch which drives the PROGRAMMED OUTPUT indicators. The information on the bi-directional data bus is then latched onto the output port at the time of the processor write strobe.

The STATUS WORD DISABLE line (SSWDSB, Pin 53 backplane) is gated to insure that no conflicts are created between the bi-directional bus drivers on the MPU and CP-A boards. This signal is controlled by the same gating that places the high order address switch values on the data bus for a front panel (address hex FF) read.

The STATUS WORD DISABLE line, Pin 53 in the backplane, is also run by the signal which puts the high order address switches onto the data bus for the port FF read instruction so that the bi-directional data bus is not being driven by the bi-directional drivers on the MPU board at the same time that the front panel is inserting the switch information on the data bus.

The RESET switch directly grounds the RESET line on the backplane which is detected by the MPU board and processed to form a RESET pulse which re-appears on the backplane as a Power On Clear.

When the RESET switch is thrown to EXTERNAL CLEAR, the switch directly grounds the EXTERNAL SWITCH line on the backplane. There is a diode between the RESET line and the EXTERNAL CLEAR line so that during a reset operation an EXTERNAL CLEAR is also generated.

## USER GUIDE

The CP-A board contains no user option jumpers or any other special connections that must be made to use the board. If the panel is mounted in the IMSAI 8080 cabinet then the power on/off switch should be connected using a separate wire to the power supply section as described in the Power Supply documentation. If the panel board is not going to be mounted in the cabinet, then the power switch should not be connected. In this case, the power switch on the inside or on the back of the cabinet would be used for controlling power to the IMSAI 8080.

Panel installation requires a backplane slot. The panel may be plugged directly into the front slot of the mother board (with the mounting screws from the acrylic face plate assembly extending through the metal panel immediately behind and secured with lock washers and nuts) or plugged on an extender card into any location in the back plane.\* The 16 pin DIP plug on the end of the flat cable must also be inserted into the corresponding socket in the upper right hand corner of the MPU-A board.

Only one front panel should be plugged into the bus at any time to avoid conflicts between multiple driving sources on the same signal lines for some of the control lines between the front panel and the system bus. The front panel is now ready to operate.

The 16 ADDRESS-DATA and ADDRESS-PROGRAMMER INPUT switches are 2 position paddle switches and represent a 0 in the down position and a 1 in the up position. The switches are provided in two colors, and can be arranged either in color groups of four to assist programming in hexadecimal or color groups of 3, 3 and 2, to aid in octal programming.

The low order byte of address switches, serve to enter into memory either data or program instructions. These switches are labeled ADDRESS-DATA 0 through 7. Each byte of data or program that is to be entered from the front panel is set into these switches after the appropriate address has been selected and entered. The switch

-----  
\*Switches should be included whether the front panel is mounted in the front of the cabinet or not.

positions are not indicated on the indicator lights until the information is deposited in memory. At that time the information from these switches appears on the data bus. The high order byte of address switches is labeled ADDRESS-PROGRAMMED INPUT and these switches can be read by the program as input port position hex FF or octal 377. The additional labels 0 through 7 are provided above these switches to assist in interpreting the switch positions when being used as an input port. The position of these switches does not appear in the indicators until the input instruction from position FF is executed, during which execution time the switch positions appear on the data bus as it is being read into the 8080 processor.

The six control switches are grouped at the right end of the panel. They are center-off two-position spring-return switches with the exception of the POWER ON/OFF switch, which is a rocker type to eliminate accidental powerdowns. The function switches are provided in alternating colors for easy identification and to reduce operator error.

The RUN/STOP switch controls program execution. When the switch is pushed to the RUN position, a control signal is sent to the processor board and enables it to start or continue executing program instruction beginning in the location indicated at that time in the address bus lights. When the address switch is depressed to the STOP position, this enable signal is removed from the processor board at the beginning of the next instruction cycle so that the processor will stop executing during the fetch cycle for that following instruction.

When the processor is enabled to RUN, the RUN light above the RUN/STOP switch will be lit. When the processor has been stopped, the WAIT light to the right of the RUN light will be lit. During normal operation, the RUN light will be on full and the WAIT light will be on partially, the exact amount depending on how many wait cycles are required by the memory and peripheral devices being run by the processor at the moment.

The front panel must be holding the processor in the stopped condition for the SINGLE STEP switch, the DEPOSIT/DEPOSIT NEXT switch, or the EXAMINE/EXAMINE NEXT switch to operate.

The EXAMINE/EXAMINE NEXT switch provides the facility for observing what is stored in memory in any location or for setting the program counter to any desired location to initialize program execution there.

When examining the contents of a location in memory, the 16 address switches are used to enter the 16 bit address. This 16 bit address is normally said to be divided into two 8-bit sections labeled high order and low order. The high order address is on the left hand side of the panel, and the low order address is in the center. The low order byte contains bits 0 through 7 and the high order byte contains bits 8 through 15. When only a small amount of memory is being used the high order bits are normally 0 and the switches must be in down position, unless the address jumper selection on the memory board is wired otherwise.

When the EXAMINE switch is actuated, the processor jumps to the address location set in the 16 address switches and is stopped during the fetch cycle out of that memory location. At that time, the address bus indicators will show the address set in the 16 address switches and the data bus indicators will show the contents of that memory location. Any additional locations in memory may be observed by setting the 16 address switches to that desired address and actuating the EXAMINE switch again. When the EXAMINE NEXT switch is actuated, the address shown in the address bus indicators is incremented by 1 and the contents of that following memory location are displayed on the data bus lights. Thus, a program or data would normally be checked by setting the first address in the address switches and actuating the EXAMINE switch to see the first byte, and thereafter actuating the EXAMINE NEXT position to observe each succeeding byte of data or program.

The DEPOSIT/DEPOSIT NEXT switch is similar in its operation but provides for changing the data or program stored in the memory. When the switch is actuated to the DEPOSIT position, the values of the lower address byte switches, that is, bits 0 through 7 labeled Address-Data, are deposited into the address currently being indicated on the 16 address bus indicators. After the DEPOSIT switch is actuated, the data will appear on the data bus indicators. If the data was incorrect because the switches were set wrong, the switches can be changed, the DEPOSIT switch actuated again, and the new values will be deposited to memory in that same location.

When the DEPOSIT NEXT position is actuated, the address currently appearing in the 16 address bus indicators is first incremented by 1 and the data entered into the ADDRESS/DATA switches is deposited in that following location and will appear in the data bus. The DEPOSIT NEXT position functions exactly the same as depressing EXAMINE NEXT to increment the address bus by 1 and then actuating DEPOSIT to deposit the ADDRESS-DATA switch positions into that location.

When the processor is stopped, instructions may be executed one at a time through the use of the SINGLE STEP switch to the right of the RUN/STOP switch. If this switch is depressed or raised, the processor board is permitted to run one instruction, and it will stop when it is in the fetch cycle in the following instruction. Thus repeated operations of this switch permit the programmer to step through his program one instruction at a time and follow what the machine is doing, noticing on the data bus what the fetched instruction is, and on the address bus the location from which that instruction is being fetched. For instructions requiring multiple memory accesses, for instance those with an address following in the second or third byte, each operation of the SINGLE STEP switch advances through only one part of the instruction. Thus, each byte of the instruction being read in and each byte of data being read in or out may be observed on the panel.

The RESET/EXTERNAL CLEAR switch provides the system reset functions. When depressed to the EXTERNAL CLEAR position the CLEAR signal is given to all external input/output interface cards which are wired to be reset by this signal. When raised to the RESET position, the 8080 processor is reset. This sets the program counter to location 0 and then returns control to the processor. If the front panel is permitting the processor to run when the RESET switch is actuated, upon release of the RESET switch the processor continues execution starting at position 0. If the front panel was holding the processor in a stopped state, during the time the RESET switch was actuated, then the program counter will be set to 0. When the RESET switch is released, the processor will remain stopped and will be positioned at memory location 0.

The 8-BIT PROGRAMMED OUTPUT INDICATOR lights can be controlled by the program through the use of the output instruction to port location hex FF or octal 377. When 0 bits are output into this port, the indicator lights will be turned on and when 1 bits are output into this port, indicator lights will be turned off.

The STATUS BYTE INDICATOR LIGHTS display the condition of the status byte during the execution of that instruction. The 8 status bits included in the status byte are the Memory Read Bit, the Input Instruction Bit, the Instruction Fetch Bit M1, the Output Instruction Bit, the Halt/Acknowledge Bit, the Stack Operation Bit, the Write Output Complement Bit, and the Interrupt Acknowledge Bit. In normal front panel operation, whenever the machine is stopped and the EXAMINE, DEPOSIT, EXAMINE NEXT or DEPOSIT NEXT switches are being used, the MEMORY READ, the M1 INSTRUCTION FETCH, and the WRITE OUT COMPLEMENT STATUS lights should be on.

While single-stepping through a program, either these or other status lights will be on as appropriate to the instruction function being executed at that moment.



For a more complete description of the functions of the status bits, reference should be made to the Intel 8080 Micro Computer Systems User's Manual. The INTERRUPT ENABLED indicator is turned on whenever the interrupts are enabled into the 8080 processor by the INTERRUPT ENABLE INSTRUCTION. This light is turned off either by an interrupt occurring and the processor acknowledging it, or by the instruction to disable interrupts. The HOLD indicator light is lit whenever a special-purpose input/output card is holding the processor so as to gain direct access to the memory on the system bus.

MPU-A

FUNCTIONAL DESCRIPTION

The MPU-A board is the processor board for the IMSAI 8080 Microcomputer System. It is designed using the Intel 8080 micro-processor chip. The bus arrangement and board connector has been chosen to be 100% compatible with the MITS Altair M8800 Microcomputer system so that all boards are 100% interchangeable between the Altair system and the IMSAI 8080 system.

Every effort has been made to keep the design simple and straight-forward to maximize reliability and ease of maintenance. MSI and LSI are used where appropriate, and discrete components are held to a minimum for greater circuit reliability and ease of assembly.

The 8224 clock driver chip and an 18 Megahertz crystal are used to generate the 2-phase, 2 Megahertz non-overlapping clock for the 8080A. An 8212 is used as a latch for the status signals and two 8216 tri-state bi-directional bus drivers are used to interface the 8080A with the IMSAI 8080 input and output data buses. All other address, status, and control lines are driven by tri-state bus drivers.

Unregulated +16, -16, +8 volts, and ground must be supplied to the bus. On-board regulation is used to arrive at the power supply levels needed to run the chips. Integrated circuit power regulators with overload protection are used. The board is supplied with ample bypass filtering using both disc ceramic and tantalum capacitors.

The board connector is a 100 pin edge connector on .125 inch centers 50 pins on each side. Dimensions are 5 inches by 10 inches, using 2 sided glass reinforced epoxy laminate, with plated feed through-holes to eliminate the need for any circuit jumpers. The contact fingers are gold-plated over nickel for reliable contact and long life. All other circuitry is tin-lead plated for better appearance and more reliable solder connections.

Power-on reset is included on this board along with pull up resistors for all inputs required so that with the front panel removed from the IMSAI 8080 machine, the power-on reset will start the program at position 0 out of a ROM. All other necessary conditions are met so that the system will run without the front panel attached, for use in dedicated controller applications where no operator-processor interaction is desired.

## THEORY OF OPERATION

The IMSAI MPU-A board is structured around the Intel 8080A microprocessor chip, and much of the MPU-A board is wired to support the 8080A device. The MPU-A board provides interfacing between the 8080A chip and the data and address busses, clock and synchronization signals, and the voltage regulation necessary for the 8080A and other chips. The internal functioning of the 8080A is thoroughly described in the Intel 8080 Microcomputer System User's Manual. Reference should be made to this manual for information concerning the operation and use of the 8080A.

The address lines from the 8080A drive the address bus on the back plane through 8T97 tri-state buffer drivers. These drivers may be disabled through the ADDRESS DISABLE line on pin 22 of the back plane. Intel 8216 bi-directional bus drivers connect the 8080's bi-directional data bus to the back plane's dual uni-directional DATA IN and DATA OUT busses. The direction of data transmission is determined by the DIRECTION ENABLE line. The DIRECTION ENABLE line is in turn controlled by the front panel and the processor status signals DATA BUS IN and HALT ACKNOWLEDGE. The 8216 can be disabled by the DATA OUT DISABLE line on pin 23 of the back plane.

The 8080A's bi-directional data bus is also connected to the data bus socket and the 8212 status byte latch. The data bus socket is used to connect the front panel to the bi-directional bus, while the 8212 latch transfers the status byte to the back plane via 8T97 drivers. These drivers are disabled by the STATUS DISABLE line on pin 18 of the back plane. The 8212 is latched up by the STATUS STROBE signal of the 8224 clock chip to store the status information for each instruction cycle.

One K pullup resistors to +5 volts are connected to all the bi-directional bus lines to ensure that during the time the bus is not driven, the 8080A reads all 1's.

The 8224 clock chip and crystal oscillator, provide the two-phase non-overlapping 2 megacycle system clock for the 8080A. These clocks are also driven onto the back plane through 8T97 tri-state buffered drivers.

The CLOCK line on the back plane is driven from the TTL Phase II clock line through a delay so that the phase relation of the clock signal to the Phase II and Phase I back plane signals, is nearly identical to that produced by the MITS Altair 8800 system. Six sections of a 7404 are used for this delay to provide greater simplicity and higher reliability than a one-shot. The 8224 chip also provides the power-on reset function through use of a 4.7K resistor and 33 uf capacitor connected to the reset input of the 8224. The power-on reset is applied to the 8080A and is applied to the POWER ON CLEAR line, pin 99 on the back plane.

The two BACK PLANE READY signals are ANDed and connected to the 8224 for synchronization with the Phase II clock before being connected to the 8080A chip. The INTERRUPT line is connected directly to the 8080A, while the HOLD REQUEST line is synchronized with the Phase II clock and then connected to the 8080A.

The six processor status signals (SYNC WRITE, STROBE DATA BIT IN, INTERRUPT ENABLED, HOLD ACKNOWLEDGED, and WAIT ACKNOWLEDGE) are all driven onto the back plane through 8T97 tri-state buffered drivers. These drivers may be disabled by the CONTROL DISABLE line, pin 19 on the back plane.

The +5 volts is regulated from the +8 volts by a 7805 integrated circuit regulator, while the -5 volts is regulated by a 5 volt zener and a 470 ohm resistor from the 16 volt bus. The +12 volts is regulated by a 12 volt Zener and connected to the +16 volt line by two 82 ohm  $\frac{1}{2}$  watt resistors in parallel. All voltages are filtered with .33 microfarad tantalum and disc ceramic capacitors.

IMSAI 8080 System  
General Assembly and  
Test Instructions

- d) 2 or 3 #14 or #12 gauge wires from ground plane to ground bus on Mother board.
- e) 2 #18 gauge wires from External Switch pads to power switch on CP-A or on back panel.
- f) 2 wires (#18 or #20 gauge) from switched AC pads to fan (if fan installed) install insulated tubing over fan terminals.
- g) 3 wires from power cord to terminals W, G, and B on PS-C. Make sure the power cord wire colors match the label on the panel.

#### INSTALLATION OF CP-A PANEL AND CONNECTION TO POWER SUPPLY

Plug the completed CP-A panel into the front 100 pin connector on the Mother board. Install the eight Allen head screws into the PEM nuts on the sheet metal front frame. Solder the two #18 gauge wires from the External Switch pads on the Power Supply assembly to the power switch pads on the CP-A panel. Provide as much clearance as possible between the connections on the CP-A board and the sheet metal front frame. Be careful not to damage the acrylic panels with the soldering iron.

#### CHECK OUT OF POWER SUPPLY

Before plugging in circuit boards except the CP-A board, the unit should be plugged into the AC power supply and the power supply turned on by depressing the front panel rocker switch. The voltages at the outputs should then be measured (any DC volt meter with a full scale voltage of 20 to 50 volts will do) and the voltages should read approximately 18 volts on the +18 and -18 volt outputs, and 10 volts on the +8 volt output. If the voltage does not come to these values, a check should be made that the positive and negative terminals of the capacitors are connected properly and the diodes are mounted properly. If there is a problem with any of these items a wiring error has probably been made and the wiring should be rechecked carefully. If the wiring is checked and no error is found, assistance should be sought from a person knowledgeable in electronics or from the factory.

When the voltage of the capacitors has been checked out to be satisfactory, the unit may be turned off. A 10 minute wait will permit the capacitors to discharge. While there is considerable energy stored in the power supply filter capacitors when they are fully charged, the voltage levels are not high enough to present a danger. Some care should be taken, however, not to discharge the capacitors by shorting them with a tool or other metallic object.

IMSAI 8080 System  
General Assembly and  
Test Instructions

With the Power Supply checked out and operating properly, the rest of the system is ready to be tested. The MPU board should be inserted in the slot behind the front panel with the flat cable inserted into the socket in the upper right hand corner of the MPU board before the board is fully seated.

The memory board should then be inserted in the third slot. While it is not necessary that the first memory board be addressed beginning at position 0, it is normally expected and the rest of this section will assume that the memory board jumpers were wired according to the directions in the User Guide section of the RAM-4A board for addressing the board at 0.

The slots in the Mother board are not unique and if a larger version (e.g., 22 slot) was ordered with more edge connectors, the boards need not be plugged into the second and third slot as directed but may be plugged into any slots.

#### SYSTEM FUNCTIONAL TEST

When the boards are installed, the machine is ready to test. Turn the power on with the front panel rocker switch and depress the RUN/STOP switch momentarily to STOP position and release. The WAIT light should be on and the RUN and HOLD lights should be off, with the other lights in various states at this time. Raise the RESET switch momentarily to the RESET position and release. All the lights on the bottom row in the ADDRESS BUS section should be indicating that the program counter is set to location 0. The WAIT light should still be on with the RUN and HOLD lights off. The DATA BUS lights may show various random bits on and the STATUS byte should have three lights on: MEMR, M1, and W0. With all 16 ADDRESS switches in the down or 0 position, the EXAMINE/EXAMINE NEXT switch

General Assembly and  
Test Instructions

should be raised momentarily to the EXAMINE position and released. Check that the lights after this operation are exactly the same as described for after the RESET switch was operated.

The machine is now ready to enter a small test program. For complete description of program operation in computers, read An Introduction To Microputers. For the initial machine test, the following program should be entered:

## TEST PROGRAM 1

ADDRESS	HEX	BINARY	OCTAL	
0	DB	1101 1011	333	INPUT
1	FF	1111 1111	377	ADDRESS
2	D3	1101 0011	323	OUTPUT
3	FF	1111 1111	377	ADDRESS
4	C3	1100 0011	303	JUMP
5	00	0000 0000	000	LOW ADDRESS
6	00	0000 0000	000	HIGH ADDRESS

## TEST PROGRAM 2

ADDRESS	HEX	BINARY	OCTAL	
0	DB	1101 1011	333	INPUT
1	FF	1111 1111	377	ADDRESS
2	2F	0010 1111	057	COMPLEMENT DATA
3	D3	1101 0011	323	OUTPUT
4	FF	1111 1111	377	ADDRESS
5	C3	1100 0011	303	JUMP
6	00	0000 0000	000	LOW ADDRESS
7	00	0000 0000	000	HIGH ADDRESS



IMSAI 8080  
General Assembly and  
Test Instructions

The address is now at 0 as indicated by the lights labelled ADDRESS BUS. Into position 0 we wish to put an input instruction.

The bit pattern for the input instruction must be set in the center group of switches labelled ADDRESS-DATA. Switches 7, 6, 4, 3, 1 and 0 should be placed in the up position. Compare these switch positions with the binary representation of the input instruction listed on the first line of test program 1. We wish now to deposit this bit pattern in memory position 0. Raise the DEPOSIT/DEPOSIT NEXT switch up momentarily to the DEPOSIT position and release. The address bus should still show 0 (no lights lit) and the data bus should now show the bit pattern set in the switches (bits 7, 6, 4, 3, 1 and 0 lit and bits 5 and 2 off).

Next, the bit pattern for the address of the input port should be written in position 1. This can be done by setting all eight ADDRESS-DATA switches up, corresponding with the address listed on line 2 of Test Program One, and the DEPOSIT/DEPOSIT NEXT switch depressed momentarily to the DEPOSIT NEXT position and released.

Now the address bus light should show position 1 (address bus light 0 on and all other address bus lights off). The data bus should show all eight lights lit corresponding to the bit pattern written here. Similarly, the next five lines of Test Program One should be set into the ADDRESS-DATA switches and deposited by operating the DEPOSIT NEXT switch, each time checking to make sure that the data bus lights correspond with the settings of the ADDRESS-DATA switches and that the address is correct indicating that no steps have been skipped or done twice.

When the last byte has been deposited in address position 6, then all 16 address switches should be returned to the 0 position (down) and the EXAMINE switch operated. This should reset the address bus lights to 0, and display the contents of the bottom word in memory on the data bus lights. (This should still be the binary pattern listed in line 1 of the Test Program). The EXAMINE NEXT switch can then be operated and the address bus lights should indicate address 1 (bit 0 on and all other bits off). The Data Bus should show the contents now of memory location one which should correspond to the second line of Test Program One listing (all ones).

The EXAMINE NEXT switch can be repeatedly operated, each time checking that the data located in the consecutive memory location corresponds exactly to the listing for Test Program One.

The EXAMINE switch can again be raised momentarily with the address switches all down, to return the machine to position 0, once it has been determined that all lines listed in Test Program One are stored correctly in the memory.

Now we can single-step through this program and watch the operation of the machine. With the machine sitting at 0 with the correct instruction on the data bus, and the MEMR, M1 and  $\overline{W}O$  lights lit in the status byte, the processor is reading the first instruction out of memory into the processor for execution. If the SINGLE-STEP switch is either depressed or raised once, it will permit the processor to complete its cycle and begin the next cycle. The address bus lights will show position 1, the data bus will show all ones corresponding to the bit pattern in the Test Program, and the status byte will show MEMORY READ and  $\overline{W}O$ . The lack of an M1 light in a status byte indicates that the processor is no longer fetching an instruction to execute, but rather this cycle it is fetching the address for the instruction which it has already stored internally.

If the SINGLE-STEP switch is operated once again, the address bus lights will all be lit. The status byte will show INP and  $\overline{W}O$  and the data bus will at first show no lights on. If one or more switches in the left hand group of eight switches is now raised or lowered, the corresponding light on the data bus indicators will turn on or off. The processor is now executing the first instruction which was an input data from address FF hex (377 octal) which is the address for the programmed input port on the front panel. By means of this instruction with this address the processor is able to read the position of the eight switches in the left hand group. (The address being read is indicated by the lights in the address bus and, on input or output instructions, the address appears in both groups of eight lights on the address bus. Thus, for this address, all the lights in the address bus are lit.)

The switches in the left hand group should be left in the position of some up and some down to provide a recognizable pattern before continuing. With the pattern left in the left hand group of switches, the single step switch can be operated once more permitting the processor to complete the execution of the input instruction, and begin the next cycle. Having completed the input instruction, the ~~next~~ cycle will be a fetch cycle during which the processor reads the next instruction to be executed, which it will find in memory address position 2. The address bus lights should now show position 2 (bit 1 on and all others off), and the data bus should indicate the bit pattern listed on line 3 of Test Program 1 for address position 2. This is the output instruction.

The Status Byte will again have MEMR, M1, and  $\overline{WO}$  lights lit and the others off. When the single step switch is operated once again, the processor is permitted to complete the cycle during which it reads in the output instruction and begin the next cycle during which it will read the address of the output device. Since it is reading this address from the next memory position, (memory position 3), the address bus will have bits one and 0 on and the others off. The Data Bus will have all lights on indicating the bit pattern we stored in memory position 3. The status bit will show MEMORY READ and WRITE OUT lights on, and the M1 light is off at this time, indicating that this is not an instruction fetch cycle, but rather it is one of the cycles required to execute the last instruction fetched-in this case, reading the address to which the data will be output. When the SINGLE STEP switch is operated once again, the processor is permitted to complete the cycle of reading the output address in and begin the next cycle which is the output operation.

The output operation looks similar to the input in that the address of the output device appears in both the upper and lower half of the Address Bus, (again in this case lighting all the lights), and the data being output appears in the Data Bus, which should show the pattern previously set in the left hand group of switches. Since the data is being output from the accumulator in the processor where it was previously stored in the input instruction, it will not be affected by moving the switches in the left hand group at this time. The Status Byte shows the MEMR light off at this time and shows the out light on indicating that the processor is executing an output instruction. The  $\overline{WO}$  light is off indicating that the processor's WRITE strobe is active. If the SINGLE STEP switch is operated once more, it will permit the processor to complete the WRITE operation and begin the next cycle. At this time, the PROGRAMMED OUTPUT lights at the top left of the panel, should be lit according to the complement of the pattern that was set in the switches. That is, for each switch that was set in the up position, the light will be out, and each switch that was set in the down position, the corresponding light will be on.

Since the processor has completed the output instruction the next cycle is used to fetch the next instruction to be executed, which it will read from memory position 4. In memory position 4 we had stored the jump instruction

which should now appear on the lights on the data bus indicators. As the SINGLE STEP switch is operated again, permitting the processor to complete the fetch of the jump instruction, and start the next cycle of executing that jump instruction, we find that the processor is reading the low half of the address from memory position 5. The status byte shows the MEMR and  $\overline{WO}$  lights lit, and the M1 light is off at this time.

If the SINGLE STEP switch is operated once again, it will be seen that the processor is reading the high address byte previously stored in memory location 6.

The next operation of the SINGLE STEP switch permits the processor to complete the execution of that jump instruction, which is instructing the processor to take its next instruction to be executed not from memory position 7 but from memory position 0 as was stored in the two bytes following the jump instruction.

The Address Bus lights should now be all off indicating that the processor is indeed fetching the next instruction from memory location 0. The Data Bus should show the pattern that we wrote in memory position 0 as the input instruction. We have now completed one cycle of the loop in Test Program 1. Further operations of the SINGLE STEP switch will let the processor step through the execution of the loop additional times and each time through the loop it is possible to set a different pattern in the left hand group of switches to be read in and later to be written out to the PROGRAMMED OUTPUT lights. The RUN/STOP switch can be momentarily raised to the RUN position and released. This will permit the processor to run at the full clock speed which will result in the loop being executed roughly 50,000 times every second. Thus, as any of the switches in the left hand group of eight are moved while the program is running, the machine reads the new position essentially instantly and displays it in the PROGRAMMED OUTPUT port above.

It may have been puzzling that the lights in the PROGRAMMED OUTPUT port seem to indicate the opposite of what might have been expected when a bit was read in as a 1 and output to the PROGRAMMED OUTPUT port. This will serve as an example of the way logic design has been affected by the appearance of large scale integration and micro-processors. While it would have been entirely possible

IMSAI 8080  
General Assembly and  
Test Instructions

and easy to provide a circuit modification such that when the data was put out as a 1 the light would be lit rather than turned off, such as addition to the circuit would have cost you more than the cost for byte of memory. The same function as the added circuit can be accomplished by adding one instruction to the loop which complements the data, that is, changes all ones to 0's and all 0's to 1's. Test Program 2 is exactly the same as Test Program 1 with the addition of one instruction between the input instruction and the output instruction, which will complement the data read in from the switches before it is output. If the machine is stopped and reset, Test Program 2 may be entered exactly the same way as Test Program 1 was and checked and then run through one or more cycles with the operation of the machine and to double-check that the program truly has been entered correctly. Then the RUN switch may be actuated to permit the loop to run at high speed.

With this change in the program, the PROGRAMMED OUTPUT port will show a light lit when the switch is positioned up to enter a 1 bit. Not only is this a less expensive way to achieve the function of causing the lights to turn on when the bit is entered as a 1, but it is a much more versatile solution since the operator can change his mind at a later date and either remove the complement instruction or change it to yet another instruction for a different result.

When single stepping through Test Program 2, the complement data instruction is seen to use up only one cycle of the processor. We are able to see it being fetched to be executed, and when the SINGLE STEP switch is operated again, we are immediately fetching the next instruction. This will be true of any instructions which operate only on data which is already stored within the processor. Additional cycles are only necessary if additional information must be read in or out of the program processor itself.

After either loop is running, the RUN/STOP switch may be depressed to STOP at any time and the operation processor will stop during the fetch of the next instruction. Due to the speed at which the processor operates, it is impossible to tell beforehand at what point in the loop the processor will be at the exact instant that the RUN/STOP switch is moved to STOP, so that the processor will stop at different places in the loop for different times when the switch is actuated.

IMSAI 8080  
General Assembly and  
Test Instructions

The switch may be raised to the RUN position starting at any point in the loop and the processor will continue to run at high speed beginning at the point. The flip-flop set by the RUN/STOP switch simply instructs the processor to wait at each cycle for a pulse which is generated by the SINGLE STEP switch to be received before executing the next cycle, and apart from waiting for this pulse, the processor executes exactly the same whether it is in the single run mode or stop mode.

The definition of a computer involves both the ability to execute in sequence of instructions which is stored inside the machine, also the ability to make a decision between on the value of data and as a result of that decision, choose between alternate possible paths of program step sequences to execute. Test programs 1 and 2 involve only the execution of a sequence of stored program steps and do not involve any decisions. Program 3 will illustrate the use of decisions in a computer program and should provide some interesting entertainment as well. It is a game program using the INPUT switches and the PROGRAMMED OUTPUT lights on the IMSAI 8080 front panel.

A pattern of lights in the PROGRAMMED OUTPUT ports is moved to the left one bit at a time, and the left hand bit which is "pushed off" the end of the programmed I/O register re-appears at the right end of the register. The rate at which the bit pattern is shifted to the left can be chosen by the binary number set in the front panel switches when the program is first started or when the machine is reset to start again. When a higher binary number is entered in these switches and program restarted, the bit pattern will shift to the left at a higher rate of speed. Initially, switches should be set for 2, that is all switches down **except** PROGRAMMED INPUT switch bit 1 on, in order that the bit pattern will be shifted slowly enough to easily see what the game program is doing. Once the program has been started, the rate at which the bit pattern is shifted to the left is not affected by any further movement of the front panel switches. From this time on, any time any one of the eight switches in the PROGRAMMED I/O group is changed, then the bit in the PROGRAMMED OUTPUT port which is directly above that switch at the moment it was moved, will change. If it was off before, it will turn on; and if it was on before it will turn off. The direction of travel of the switch is not significant--only that its position was changed. After a switch change is detected, and the light above it turned on or off as

appropriate, no further switch movements will affect the condition of any of the lights until the next shift to the left has occurred. This was done to give the switches time to stop bouncing and stay closed as the processor in this machine is quite fast enough to see the slight bouncing of the switch contact when it initially closes.

By waiting for the next data shift before recognizing any more switch changes, we are prevented from falsely interpreting a bouncing contact as a switch which was repeatedly opened and closed. The object of the game can be either to turn out all the lights in the shifting bit pattern by moving a switch when the bits are passing directly over it, or alternately to turn on all the bits in the shifting bit pattern by moving a switch when a bit which is off is directly over it. Any time the shifting bit pattern is all 0's or all 1's, no movement will be seen in the PROGRAMMED OUTPUT port but by moving any switch, one of the lights will be changed so that the motion is again apparent.

Players can compete for the shortest time to go from all 0's to all 1's, or the other way - from all 1's to all 0's. When the game has been mastered at one rotation speed, the switches can be set for a higher binary number and the system reset to cause the processor to go back to memory location 0 and begin execution of the program again, and a new switch setting will be read to result in a higher rate of rotation, which makes it harder to move a switch at the exact instant the bit desired to be changed is directly above it. If there were only single light on circulating across the output port, and the player, (in attempting to turn it off by moving the switch when the bit was directly over that switch) was too slow, then the bit will have shifted away so that it is now over the next switch to the left, not only will that bit not be turned off, but the bit behind will be turned on so that now there are two bits circulating across the register and the player is further away from achieving all bits turned off.

Knowledge of some of the internal structure of the 8080 processor will be necessary to understand the game program. The Intel data book contains complete information and functional specifications on the internal structure of the 8080 processor, but only the basic aspects of the structure need be known to understand the program operation.



Figure 1 shows the structural blocks in the processor which are important to the programmer. Central to the processor's operation is the register named the ACCUMULATOR. This register and all the others is like one eight bit position in memory or a small "blackboard" with room for only eight bits of either 1's or 0's to be written. When the input instruction was executed during programs 1 and 2, the pattern from the switches on the front panel was read into the ACCUMULATOR register, and when the OUTPUT instruction was given it was again the contents of the ACCUMULATOR which was output to the PROGRAMMED OUTPUT port on the front panel. All arithmetic is done in the ACCUMULATOR and, except for special instructions, (to permit other registers to be read to or from memory) all programmed input/output from either memory or input/out interfaces goes to and from the ACCUMULATOR. The INSTRUCTION register is another "blackboard" with room to store the address where it last read a program byte from memory so that when it finished the execution of that step, it can increment that address by one and use it to determine where to get the next instruction.

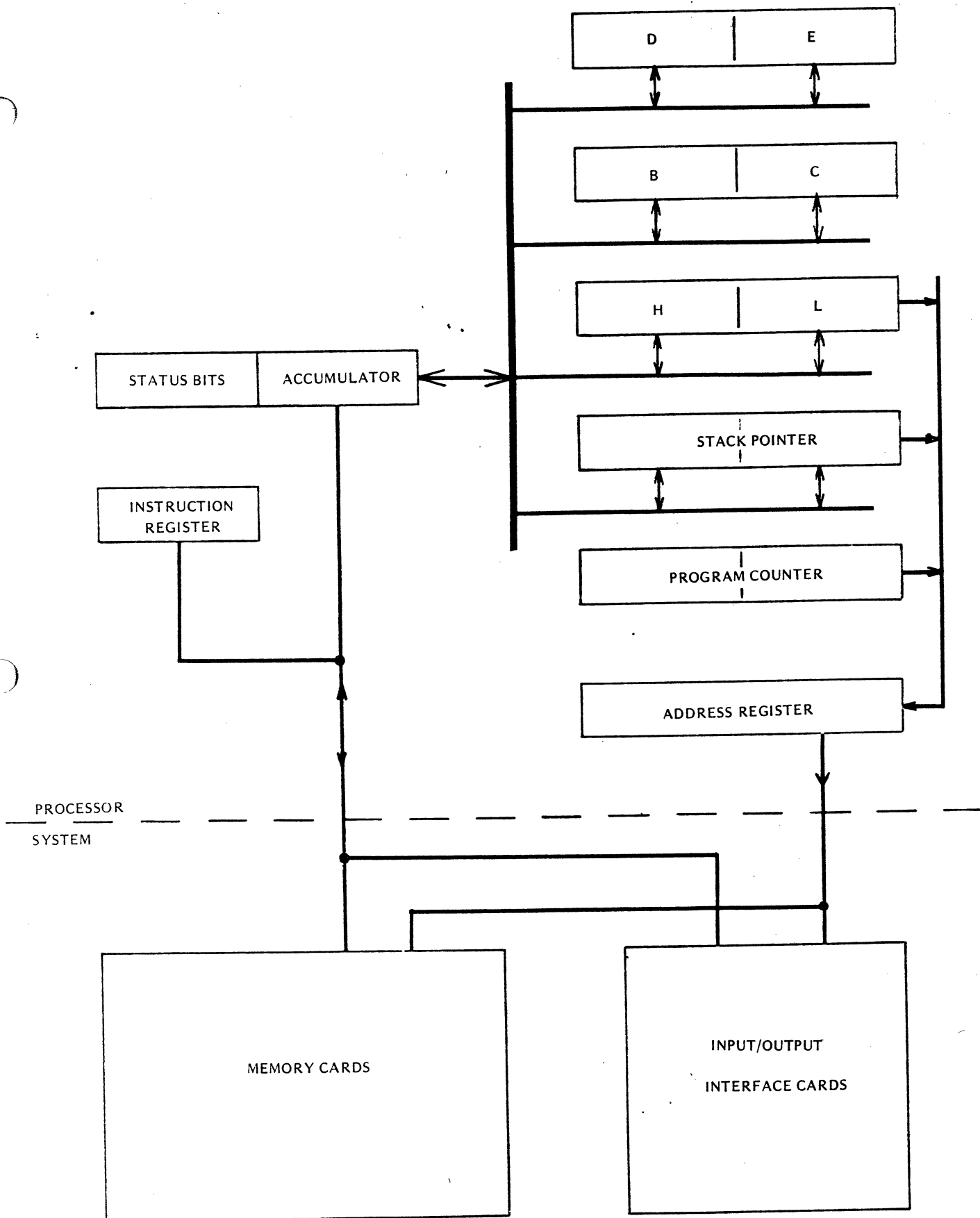
The STATUS BITS are 5 bits that are set to 1 or 0, according to the results of the last data operation performed in the ACCUMULATOR. One of the STATUS BITS or condition flags is the Z bit (zero bit) which is turned on when the last operation in the ACCUMULATOR resulted in the ACCUMULATOR being left all 0's. Otherwise, this bit is turned off. The second condition flag is the sign bit. If the most significant bit of the result of the last operation in the ACCUMULATOR has the value 1, this flag is set to 1, otherwise it is reset to 0. Three other condition flags are the sign parity and the auxiliary carry, and their functions are described in the Intel Data Book on page 4-2. The fifth condition flag is a carry flag which is turned on if the last arithmetic operation produced an overflow. An overflow is produced, for example, when two numbers are added together and their sum is too large to be contained in the register into which it is put. For instance, if the ACCUMULATOR contained eight 1's and another number was added which contained the value 6, the correct answer would be the combination of the value 5 and a bit turned on in the 9th position. Since the ACCUMULATOR has only eight positions, the carry bit would be turned on.

Some of the STATUS BITS are affected by the operations in other registers than the ACCUMULATOR. For instance the carry bit is affected by additions made in the H and L registers by using the double add instructions. Use is made of this in the game program. There are five other registers in the processor, each of which is 16 bits long, and some of which are divided in half so that operations may be done with only 1/2 at a time. The ADDRESS REGISTER is a 16 bit register over which the programmer has no control. It is simply used to output either the memory address or the input/output address necessary to execute the next cycle. The other four 16 bit registers can all be used by the programmer. There are many instructions in the Intel 8080A processor's instruction set whose function is to move data from any register to any other register, to permit arithmetic operations between a register and the ACCUMULATOR (with the result always being left in the ACCUMULATOR), and some special instructions to permit direct transfer of data from memory to a register, or vice versa.

The B, C, D, and E half registers are all general purpose registers. The H and L register pair and the STACK POINTER register pair both have special functions in addition to being usable for general purposes. The game program does not make use of these special functions.

With the basic structure of the processor in mind, we can now look at the operation of the game program. Larger programs can not be readily understood or written by working directly on the list of machine instructions, such as we did for Test Programs 1 and 2. A flow diagram is essential to quickly follow the sequence of the instructions and understand how they work together to achieve the desired result.

Figure 2 shows a flow diagram for Program 2. Each program function is briefly described in a separate box, and the flow of the execution of the program is indicated by the lines. Test Program 2 was a simple loop with no decisions so that after executing the short sequence of instructions, the flow of the program is back to the beginning of the loop to begin again. Figure 3 shows the flow diagram for the game program. Although it need not be understood to execute the game program, a thorough understanding of how this flow diagram achieves the operation of the game will be a useful step towards writing your own programs.



8080 PROCESSOR  
FIG. 1

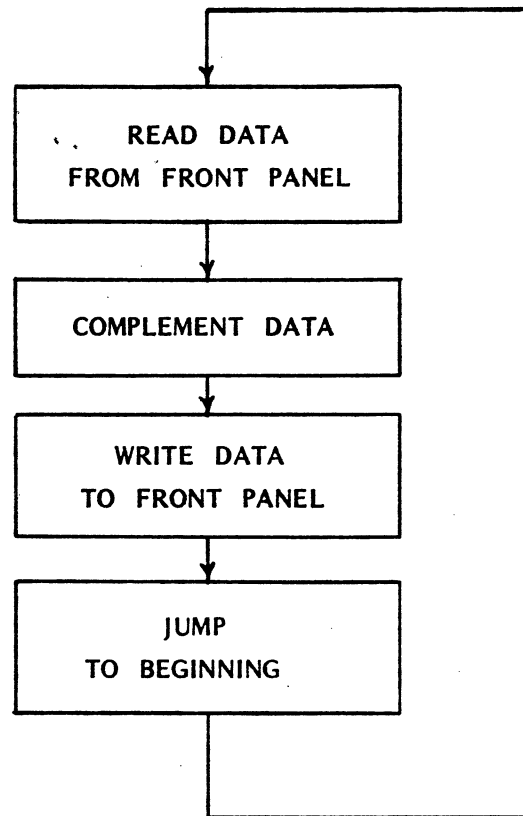
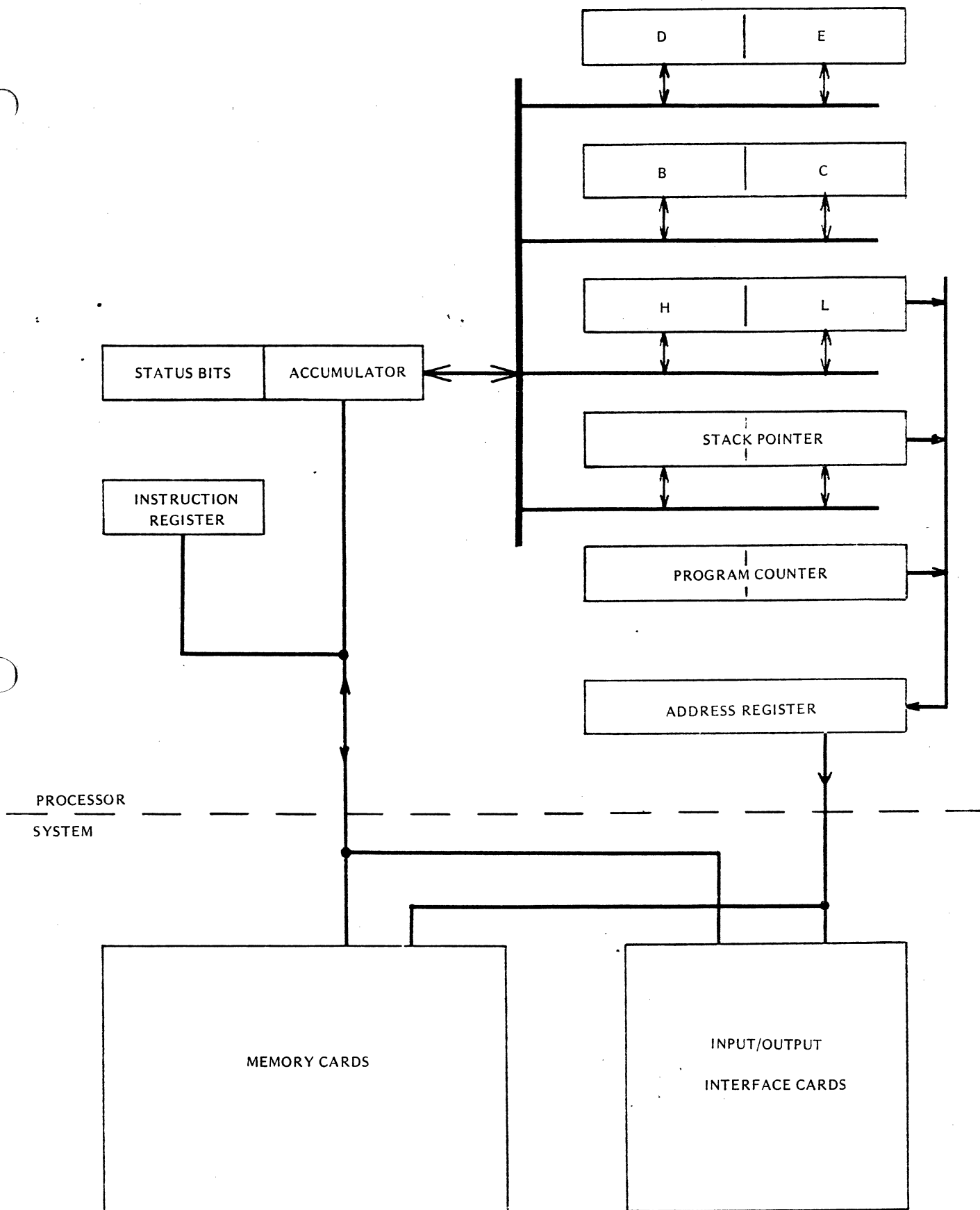


FIG. 2



8080 PROCESSOR  
FIG. 1

# GAME PROGRAM LISTING

OCTAL		HEX		MNEMONIC	DESCRIPTION
ADD.	INST.	ADD.	INST.		
000 000	257	0000	AF	XRA, A	Exclusive OR A to itself (put zero in A)
001	147	01	67	MOV H, A	Move A to H (put zero in H)
002	333	02	DB	INP	Input data
003	377	03	FF		from front panel switches
004	157	04	6F	MOV L, A	Move A to L
005	371	05	F9	SPHL	Put H&L reg. into SP
006	257	06	AF	XRA, A	Exclusive OR A to itself (put zero in A)
007	201	07	81	ADD C	Put C in A, affecting flag bits
010	302	08	C2	JNZ	Jump if not zero
011	023	09	13		(skip switch test for debounce after a switch change)
012	000	0A	00		
					IF NORMAL, CONTINUE
013	123	0B	53	MOV D, E	Move E to D
014	333	0C	DB	INP	Input data
015	377	0D	FF		from front panel switches
016	137	0E	5F	MOV E, A	Move A to E
017	252	0F	AA	XRA, D	Exclusive OR D to A
020	302	10	C2	JNZ	Jump if result not all O's
021	041	11	21		(change display if switch position changed from last time)
022	000	12	00		
					IF SWITCHES UNCHANGED, CONTINUE
023	071	13	39	DAD SP	Add SP to HL
024	322	14	D2	JNC	Jump if no carry results
025	006	15	06		(return to read switch loop if no carry yet)
026	000	16	00		
					IF CARRY, CONTINUE
027	170	17	78	MOV A, B	Move B to A
030	007	18	07	RLC	Rotate left 1
031	107	19	47	MOV B, A	Store A in B
032	323	1A	D3	OUT	Output A
033	377	1B	FF		in front panel lights
034	257	1C	AF	XRA, A	Exclusive OR A to itself (put zero in A)
035	117	1D	4F	MOV C, A	Move A to C (Reset debounce indicator)

GAME PROGRAM LISTING (CONT.)

OCTAL		HEX		MNEMONIC	DESCRIPTION
ADD.	INST.	ADD.	INST.		
036	303	1E	C3	JMP	Jump
037	006	1F	06		(to read loop)
040	000	20	00		
CHANGE DISPLAY IF SWITCH DIFFERENT					
041	250	21	A8	XRA, B	Exclusive OR B with A
042	107	22	47	MOV B, A	Store A in B
043	323	23	D3	OUT	Output A
044	377	24	FF		in front panel lights
045	257	25	AF	XRA, A	Exclusive OR a with itself A (put zero in A)
046	147	26	67	MOV H, A	Move A to H (set counter to insure enough delay for debounce)
047	057	27	2F	CMA	Complement A (to all I's)
050	117	28	4F	MOV C, A	Move A to C (set C to debounce)
051	303	29	C3	JMP	Jump
052	006	2A	06		(to read loop)
053	000	2B	00		

NOTE:

Exclusive OR of two switch patterns results in I's in positions which were changed, with all 0's elsewhere.

B= DISPLAY BYTE STORAGE

C=SWITCH DEBOUNCE INDICATOR

I=DEBOUNCE 0=NORMAL OPERATION

D=LAST SWITCH SETTINGS

E=CURRENT SWITCH SETTINGS

H,L=DELAY COUNTER

SP=INCREMENT FOR DELAY COUNTER